People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Abdelhamid Mehri – Constantine 2

Faculty of New Technologies of Information and Communication (NTIC)

Department of Fundamental Computing and its Applications (IFA)

MASTER'S THESIS

to obtain the diploma of Master degree in Computer Science

Option: Sciences and Technologies of Information and Communication (STIC)

Cybersecurity of smart grid infratsructure communication

Realized by:
ochetati ilyes chiheb eddine
kechicheb ahmed

Under supervision of:
Salim benayoune

June 2024

بسم الله الرحمن الرحيم

Acknowledgments

(This section allows you to thank all the people who have participated in the successful development of the end of studies project, and especially when writing your thesis. This **must not exceed 1 page maximum**.)

Dedication

(In this section, you dedicate this thesis to important people for you. This **should not also exceed 1 page**.)

ملخص

(هنا تضع ملخص المذكرة باللغة العربية. يجب أن يكون الملخص موجزا وأقل من 200 كلمة. يمكن أن يحتوي الملخص على كلمات لاتينية، مثلا NTIC .)

الكلمات المفتاحية: (6 كلمات مفتاحية على الأكثر مفصولة بفواصل " ، ")

Abstract

The smart grid's reliability and security are at significant risk due to the increasing reliance on smart grid technologies, which have brought in new vulnerabilities which is cyber threats. Many traditional intrusion detection systems (IDSs) are unable to detect most of these threats because of the complexity and of smart grid data.This thesis introduces a deep learning-based network intrusion detection system (DL-NIDS) that tackles this issue

The DL-based NIDS uses an innovative architecture utilizing convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to discover deviations in smart grid network traffic. This includes normal patterns as well as those that are maleficent.

To estimate its performance, these models were trained on real life network traffic data then tested on a new set of data. In contrast to conventional IDSs, there is a dramatic improvement in both robustness and detection accuracy as shown by results. The system is also demonstrated to be capable of detecting a variety of DoS and DDoS attacks. Therefore, this research enhances the dependability and safety of such critical infrastructure systems by contributing towards better cybersecurity solutions for smart grids.

Keywords: (smart grid, cybersecurity, deep learning, denial of service attacks, Convolutional neural network , Long short-term memory)

Résumé

La fiabilité et la sécurité du réseau intelligent sont considérablement menacées en raison du recours croissant aux technologies de réseau intelligent, qui ont introduit de nouvelles vulnérabilités que sont les cybermenaces. De nombreux systèmes de détection d'intrusion (IDS) traditionnels sont incapables de détecter la plupart de ces menaces en raison de la complexité et des données des réseaux intelligents. Cette thèse présente un système de détection d'intrusion réseau basé sur l'apprentissage profond (DL-NIDS) qui s'attaque à ce problème.
Le NIDS basé sur DL utilise une architecture innovante utilisant des réseaux neuronaux convolutifs (CNN) et des réseaux neuronaux récurrents (RNN) pour découvrir les écarts dans le trafic du réseau intelligent. Cela inclut les schémas normaux ainsi que ceux qui sont maléfiques.
Pour estimer ses performances, ces modèles ont été formés sur des données réelles de trafic réseau puis testés sur un nouvel ensemble de données. Contrairement aux IDS conventionnels, les résultats montrent une amélioration considérable de la robustesse et de la précision de la détection. Le système s'est également révélé capable de détecter diverses attaques DoS et DDoS. Par conséquent, cette recherche améliore la fiabilité et la sécurité de ces systèmes d'infrastructures critiques en contribuant à de meilleures solutions de cybersécurité pour les réseaux intelligents.

Keywords: (grille intelligente, cybersécurité, apprentissage profond, attaques par déni de service, réseau neuronal convolutif, mémoire à long terme)

Table of Contents

List of Figures

List of Tables

List of Algorithms

General Introduction

Project Background

Growing reliance on smart grid technology has necessitate the extensive augmentation of the communications infrastructure, which has inherently generated new points of vulnerability and exposure to cyber threats. A smart grid communication infrastructure is a sophisticated arrangement of devices, systems, and protocols that allows the secure and efficient delivery of electricity from generation to consumption. But as transmission interconnections expanded over time, so have the cyber threats that could — were an adversary so inclined — weaken the grid's credibility and robustness.

Problem

Problem To Be Addressed: The episodic development of this thesis are the need for effective cybersecurity mechanisms to secure the smart grid communication infrastructure from cyber threats. The greater use of smart grid technology has also increased the vulnerability to cyber attacks, whose consequences can be the disruption of the grid being attacked. The smart grid communication infrastructure is a prime target for cyber attacks, which can result in catastrophic outcomes such as blackouts, privacy breaches and even destruction of life and property.

Proposed Solutions

This thesis is aimed at developing a deep learning-based intrusion detection system using LSTM and CNN with the objective of enhancing the accuracy and efficiency of intrusion detection in smart grid communication infrastructure. The goals of this research include:

- Designing and implementing a deep learning intrusion detection system for security threats (DoS and DDoS) facing smart grid communication networks with LSTM and CNN methods.
- Evaluating the proposed system's performance using evaluation metrics such as accuracy, precision, recall, and F1-score.

Document Plan

This thesis is organized as follows: In the first chapter, we provide an overview and the state of the art of the smart grid communication infrastructure and the importance of cybersecurity. In the second chapter. In the second chapter, we describe the proposed system and its components. And we present the experimental results and evaluation of the proposed system. Finally, in the conclusion, we conclude the thesis and discuss the implications of the results.

Chapter 1

State of the Art

Introduction

Electricity grids, commonly referred to as "grids," play a vital role in modern energy infrastructure. They facilitate power generation, transmission, distribution, and control. Over time, grids have evolved from localized systems to interconnected networks, adapting to meet increasing demands and technological advancements. These grids contribute significantly to economic and societal progress.

Amidst dynamic changes in the energy landscape, the emergence of the "smart grid" presents transformative possibilities. Leveraging data, automation, and connectivity, smart grids enhance energy management and promote sustainability. In this chapter, we delve into the evolution of grid systems and explore the challenges and opportunities associated with smart grid technology, shaping the future of energy.

1.1 Definition smart grid

The Smart Grid is a comprehensive electrical network that employs cutting-edge communication technologies, computational intelligence, and cybersecurity protocols throughout the entire process of generating, transmitting, distributing, and consuming electricity. Its objective is to establish a system that is environmentally friendly, secure, dependable, adaptable, energy-efficient, and environmentally sustainable. While the ultimate vision of the Smart Grid is ambitious, its practical implementation demands careful evaluation of costs, rigorous testing, and validation. Introducing new functionalities can occur autonomously, with each necessitating justification and a reasonable return on investment. The compatibility of open systems facilitates smooth integration into the Smart Grid once the technologies have been validated [7]. The National Institute of Standards and Technology (NIST), operating within the U.S. Department of Commerce, has classified the smart grid into seven distinct domains, as illustrated in Figure 1.1. A concise overview of these domains and their stakeholders is provided in Table 1.1.[1]

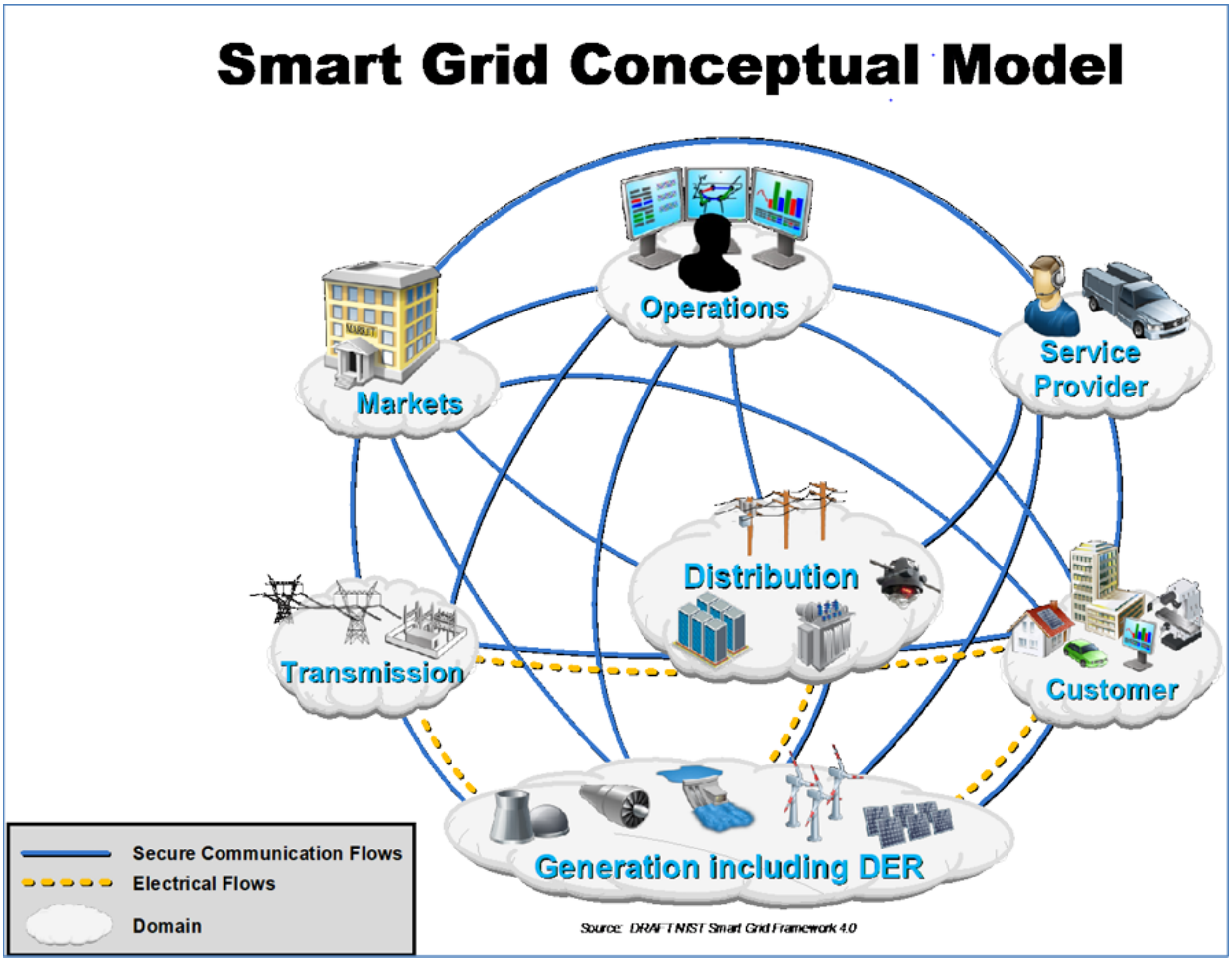

Figure 1.1: The NIST Conceptual Model for SG [1]

Domain	Roles/Services in the Domain
1 Customer	The end users of electricity. May also generate, store, and manage the use of energy. Traditionally, three customer types are discussed, each with its own sub-domain: residential, commercial, and industrial.
2 Markets	The facilitators and participants in electricity markets and other economic mechanisms used to drive action and optimize system outcomes.
3 Service Provider	The organizations providing services to electrical customers and to utilities.
4 Operations	The managers of the movement of electricity.
5 Generation Including DER	The producers of electricity. May also store energy for later distribution. This domain includes traditional generation sources and distributed energy resources (DER).
6 Transmission	The carriers of high voltage electricity over long distances. May also store and generate electricity.
7 Distribution	The distributors of electricity to and from customers. May also store and generate electricity.

Table 1.1: Domains and their associated roles/services [1]

1.2 Smart grid attributes

Many smart grid advocates cite some or all of its following attributes as representative of its promise:

- **Efficiency:** Capable of meeting growing consumer demand without the need for additional infrastructure.
- **Flexibility:** Able to accept energy from various sources, including solar and wind, with the same ease as traditional fuels like coal and natural gas. It can integrate new technologies, such as energy storage, as they become commercially viable.
- **Empowering:** Facilitating real-time communication between consumers and utility providers, allowing consumers to adjust their energy usage based on factors like price and environmental concerns.
- **Opportunistic:** Creating new markets and opportunities by leveraging plug-and-play innovations whenever suitable.
- **Focus on Quality:** Able to deliver reliable power without disruptions, ensuring the smooth operation of digital technologies crucial to our economy.
- **Resilience:** Increasingly resistant to cyber attacks and natural disasters through decentralization and the implementation of smart grid security measures.
- **Environmental Sustainability:** Contributing to the mitigation of climate change

and offering a viable path towards reducing the environmental impact of electricity generation. [8]

1.3 Differences between Traditional grid and Smart grid

Table 1.2 offers a thorough comparison of the conventional power grid with the smart grid. In contrast to the traditional grid where customers play a passive role, the smart grid actively engages them through bi-directional communication technologies. For instance, rooftop photovoltaic solar panels produce electricity during the day, enabling customers to sell surplus energy back to the grid. At night, these panels continue to power home appliances as usual. Moreover, the smart grid incorporates innovative technologies like distributed generation, electric vehicle charging and discharging, and Flexible Alternating Current Transmission Systems (FACTS) to improve energy distribution and management.[9]

Table 1.2: Comparison between conventional grid and smart grid [6]

Aspects	Conventional Grid	Smart Grid
Interaction between Grid and Customers	Customers passively accept service from grid	Customers participation on the grid action
Renewable Energy Integration	Having trouble with renewable penetration	Integration with renewable resources enhancement
Options for Customers	No choice for customer, monopoly market	With digital market trading, PHEV, introduce bids and competition, more choice for customer
Options on Power Quality (PQ)	No choice on power quality, no price plan options for consumers	Power quality levels for different consumers
System Operation	Ageing power assets, no efficient operation	Assets operating optimization, less power loss
Protection	Only rely on protection devices, fault detect manually	Have capability of self-healing, less damage affected by fault
Reliability and Security	Susceptible to physical and cyber attack	More reliable for national security and human safety

1.4 Major systems

1.4.1 Smart infrastructure system

The smart infrastructure system consists of three main components: the smart energy subsystem, the smart information subsystem, and the smart communication subsystem. Within the smart energy subsystem, activities such as electricity generation, transmission, distribution, and consumption are integrated. The smart information subsystem includes functions like smart metering and advanced monitoring and management of the smart grid network. The smart communication subsystem facilitates wired and wireless communication between networks, devices, and applications to establish connectivity throughout the network [10].

1.4.2 Smart management system

The smart grid's intelligent management system offers advanced services in monitoring and control. As innovative management, monitoring, and control applications evolve, smart grid technology becomes more sophisticated, contributing actively to the advancement of a sustainable power system. Within the smart management system are functions such as enhancing energy efficiency, balancing supply and demand, controlling emissions, reducing operational costs, and maximizing utility. This system utilizes modern machine learning and optimization tools to create a resilient and efficient smart management framework [10].

1.4.3 smart protection systems

The smart protection system within the smart grid offers services related to reliability, safeguarding against failures, and ensuring security and privacy. By incorporating advanced protection devices and monitoring tools, the system enhances the reliability, security, and privacy of the network. Alongside smart infrastructure planning, efficient management, and intelligent protection systems play a role in managing operations effectively, protecting against failures, and addressing cybersecurity and privacy concerns within the network. Figure 1.2 illustrates a typical technological framework of the smart grid [10].

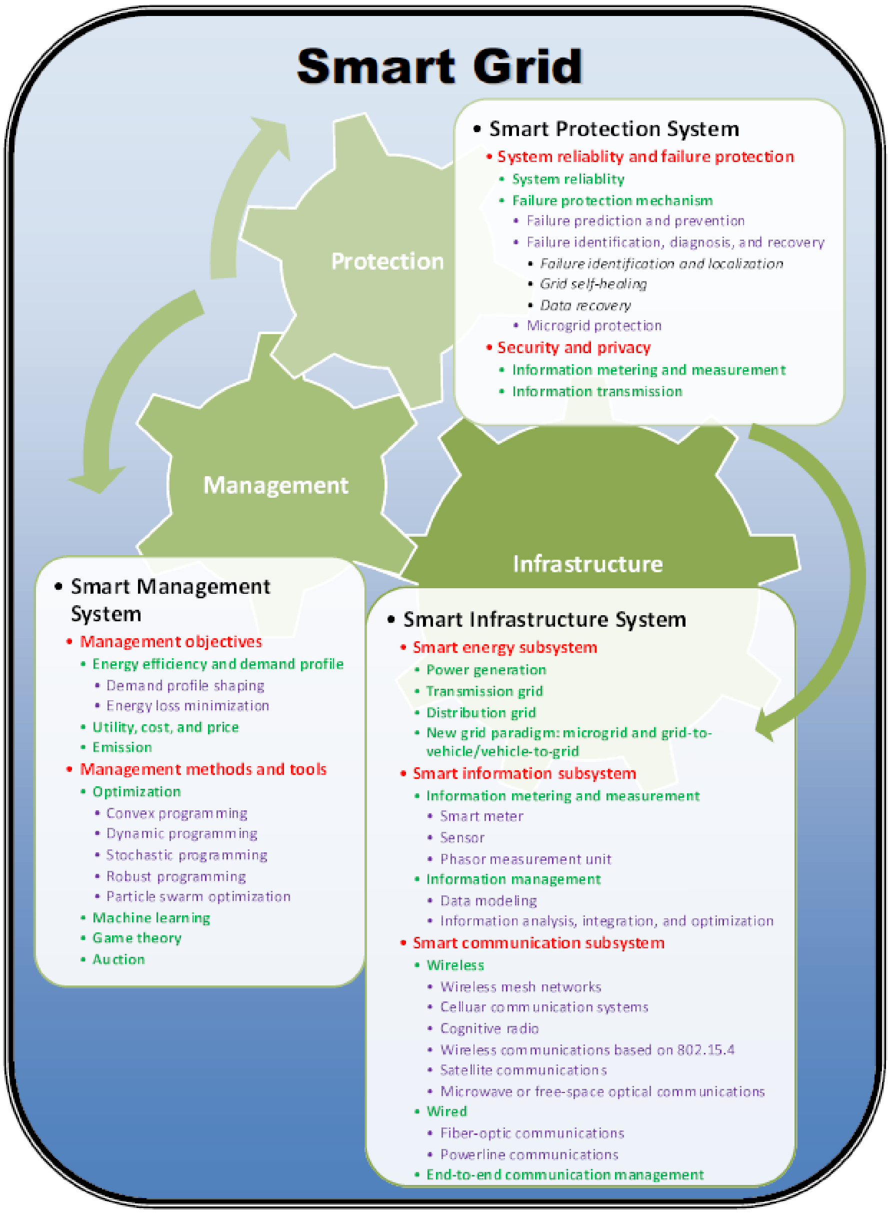

Figure 1.2: Classification of the Smart Infrastructure System, the Smart Management System, and the Smart Protection System [2]

1.5 Smart Grid Technologies

A smart grid employs a diverse array of technologies and communication networks to enhance the management of power generation, transmission, and distribution. It also provides customers with the ability to have real-time control over their energy consumption [11].

1.5.1 Major Smart Grid Technologies

1.5.1.1 Advanced Demand Forecasting

Utilizing data analytics and machine learning (ML), advanced demand forecasting techniques produce forecasting reports through autoregressive integrated moving average (ARIMA) and various statistical methods.

A crucial aspect of smart grid management, ARIMA forecasting predicts both annual electricity consumption and hourly electricity prices.

Furthermore, ARIMA forecasting serves as an extra layer of verification, aiding in the identification of cyber intrusion attempts on smart meters used to measure electricity usage for residential and commercial consumers [11].

1.5.1.2 Advanced Metering Infrastructure (AMI)

Advanced Metering Infrastructure (AMI) is a unified system comprising communication networks, data management systems, and intelligent meters designed to enhance customer service, energy efficiency, and cost management.

AMI facilitates two-way communication between customers and utilities, offering a wide array of advantages to the smart grid. These include forecasting consumption, improving revenue collection and theft detection, detecting faults and outages, measuring losses, and implementing time-based pricing [11].

1.5.1.3 Big Data

Smart grid data possesses three fundamental characteristics: high velocity, extensive volume, and diverse variety. Managing this large volume of data in a timely manner with limited resources poses a significant challenge for smart grids. This is where big data analytics becomes pivotal, offering the potential to boost asset utilization, efficiency, system reliability, and customer satisfaction.

Without big data analytics in the smart grid, the assessment of petabytes of data generated by smart grid devices would be impractical. Big data captures and analyzes unstructured data from various endpoints within a smart grid.

Moreover, big data facilitates efficient cost reduction, optimal resource distribution, and improved customer service [11].

1.5.1.4 Distributed Energy Resources (DERs)

Distributed Energy Resources (DERs) supply energy and improve local reliability, enhancing grid stability and optimizing on-site fuel utilization.

DERs encompass various technologies such as electric vehicles, solar panels, small natural gas generators, and controllable loads like electric water heaters and HVAC systems.

Efficient integration of DERs enhances grid service quality and reliability. For instance, photovoltaic systems (PVs) utilize the photovoltaic effect to convert sunlight into electricity, which is then transformed into alternating current by an inverter. The primary advantage of PV systems is reduced utility bills due to decreased reliance on grid-provided electricity [11].

1.5.1.5 Non-intrusive Load Monitoring (NILM)

Non-intrusive load monitoring (NILM), also known as non-intrusive appliance load monitoring (NIALM), discerns the specific energy consumption of households and industrial sites.

By disaggregating the total energy usage (from active appliances) into individual components and offering diagnostic insights, NILM aids in identifying energy-intensive or faulty appliances.

Moreover, consumers can optimize the timing of usage for energy-intensive appliances to minimize costs, and monitor and control energy expenses based on their power consumption [11].

1.5.1.6 Vehicle-to-Grid (V2G)

Also known as vehicle-grid integration (VGI), vehicle-to-grid (V2G) technology transfers unused power from a vehicle into the smart grid. An electric vehicle (EV) battery is a cost-efficient form of energy storage.

V2G helps balance electricity consumption spikes and reduce overload on the power grid during peak hours.

For example, V2G can feed energy (unused battery capacity) back to the power grid from an electric car's battery to improve grid stability and maximize the benefits of renewable energy [11].

1.5.2 Established and Emerging Smart Grid Communication Networks

1.5.2.1 HAN

A smart meter supplies power to household appliances via the Home Area Network (HAN), which utilizes different technologies such as Bluetooth, Wireless Ethernet, Wired Ethernet, and Zigbee. The HAN links home appliances with the smart meter, which detects power usage and transmits this information to the server for billing purposes [11].

1.5.2.2 NAN

A Neighborhood Area Network (NAN) is an external access network that links distribution automation devices and smart meters to WAN gateways such as RF (radio frequency) collectors and field devices (like Intelligent Electronic Devices (IEDs)). NAN allows for customer data collection and facilitates communication within the WAN-premise area [11].

1.5.2.3 WAN

A wide area network (WAN) uses fiber optics, 3G/LTE (Long Term Evolution)/GSM (Global System for Mobile Communication), or WiMAX (Worldwide Interoperability for Microwave Access) for communication between a smart meter, suppliers, and the utility server. A smart meter sends notifications it receives (via HAN) from the devices to the suppliers using WAN [11].

1.5.2.4 LoRaWAN

LoRa (Long Range) is a popular IoT (Internet of Things) technology known for its long-range capability and low-power wireless platform, making it well-suited for various applications including energy management, infrastructure efficiency, and disaster prevention.

Implementing smart electricity metering solutions and smart grid networks using the LoRaWAN® (Long Range Wide Area Network) protocol allows for improved understanding of power demand, efficient detection of power outages, enhanced connectivity, and identification of underperforming assets.

Additionally, LoRaWAN is globally compatible and ensures seamless transmission without interference for remote reading of heat meter consumption data [11].

1.6 Components of the Smart Grid

There are many components, but will talk about the most important ones.

1.6.1 Smart Meters

The interplay between smart meters and smart grids is depicted in Figure 1.3. From the perspective of the smart grid, the applications primarily revolve around leveraging smart meters to facilitate the coordination of various electrical devices, thereby achieving a dependable power system. Simultaneously, these applications strive to enhance the performance and efficiency of smart metering. These objectives align with the defining characteristics of smart grids, which drive the advancement of smart meter technologies.[3]

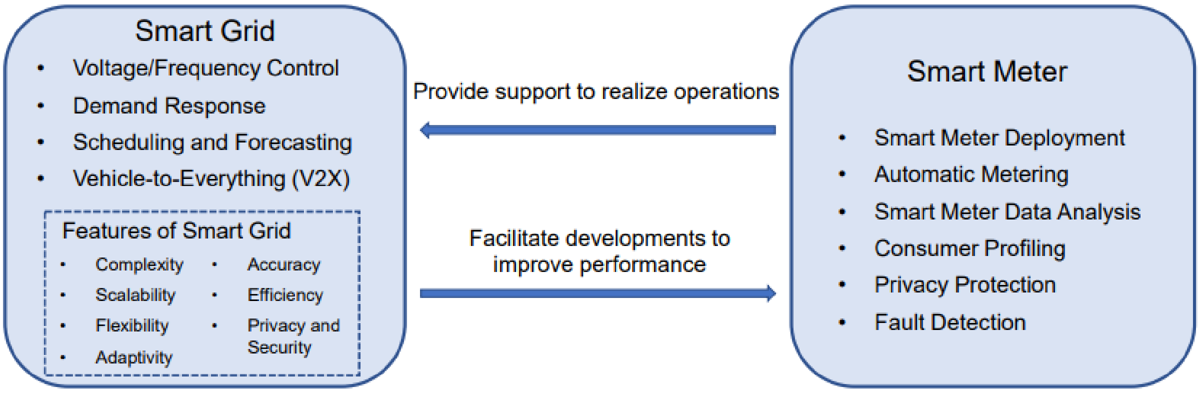

Figure 1.3: Applications from smart grid and smart meter perspectives. [3]

1.6.2 Advanced Distribution Management Systems

An ADMS is a software platform designed to support the comprehensive suite of tasks related to managing and optimizing the distribution of electricity. It encompasses functions that automate outage recovery and enhance the effectiveness of the distribution grid. These functions being developed for electric utilities include fault location, isolation, and restoration; optimization of voltage and reactive power; energy conservation through voltage reduction; management of peak demand; as well as support for microgrids and electric vehicles [12].

1.6.3 Super conducting cables

These components are utilized for transmitting electricity over extended distances and employing automated monitoring and analysis tools. These tools have the capability to identify faults independently or predict potential cable failures by analyzing real-time data, weather conditions, and the history of outages [13].

1.6.4 Circuit breakers

The circuit breaker, a safety-conscious middle-aged individual, safeguards the power system. It evaluates smart grid scenarios against safety standards, offering recommendations. Despite occasional differences, both the smart grid and circuit breakers recognize their mutual importance. Their relationship ensures the power system's safe and stable operation.

Smart Grid occasionally proposes innovative features or adjustments to the power system. However, the breaker gently dissuades excessive risk-taking, emphasizing safety. Through discussions and disputes, they strike a balance, ensuring reliable electricity for our lives [14].

1.6.5 Collector nodes

Collector nodes are pivotal in the Smart Grid, serving as points of data collection and distribution between energy suppliers and customers. They enable a two-way communication network within the grid, relaying information from customer premises to the utility control center and transmission/distribution substations. Collector nodes facilitate efficient monitoring and management of energy usage [15].

1.7 Challenges and Considerations

1.7.1 Stakeholder Engagement

At the early stages of smart grid implementations, stakeholders' negative perceptions can derail even the most beneficial project, especially when the proponents fail to pay close attention to the educational aspects. Advocates need to be able explain and clearly identify the benefits of each component of the smart grid to the customers that are the potential key to service success [8].

1.7.2 Fear of obsolesce

As many technology users (computers, smart phones, etc.) are painfully aware, the adoption of new tools can open the door to new and additional costs that may only be borne by the eventual consumer. This fear can be addressed through the development of interoperability standards and backward compatibility of technologies [8].

1.7.3 Cybersecurity

Without a shred of doubt, cybersecurity stands out as one of the foremost and intricate challenges confronting IoT devices. Sensors, devices, and networks connected to the internet are persistent targets for various online threats like probing, espionage, ransomware, theft, and potential destruction. Considering that an IoT-driven smart grid can encompass potentially millions of interconnected nodes spread across extensive geographic regions, it emerges as the most susceptible to substantial cyber assaults. Consequently, a cyber-attack on such a system would have devastating consequences, leading to significant financial losses and potentially bringing entire countries to a standstill. The diagram in Figure 1.4 illustrates the number of articles reviewed per year of publication and smart grids impacted by cyber-attacks. Hence, security stands as a major hurdle in both the deployment and operation of IoT-based smart grid networks.[16].

1.7.4 Data privacy

Privacy is a critical concern within smart grid networks, prompting significant questions about the creation of policies regarding user data privacy. These questions revolve around several key points: Who owns the customer data? How is access to and usage of customer data regulated? What measures exist to protect the privacy and security of customer data from potential risks like surveillance or illicit activities? Is it permissible to sell or transfer customer data, and under what circumstances and for whose benefit? In areas with retail

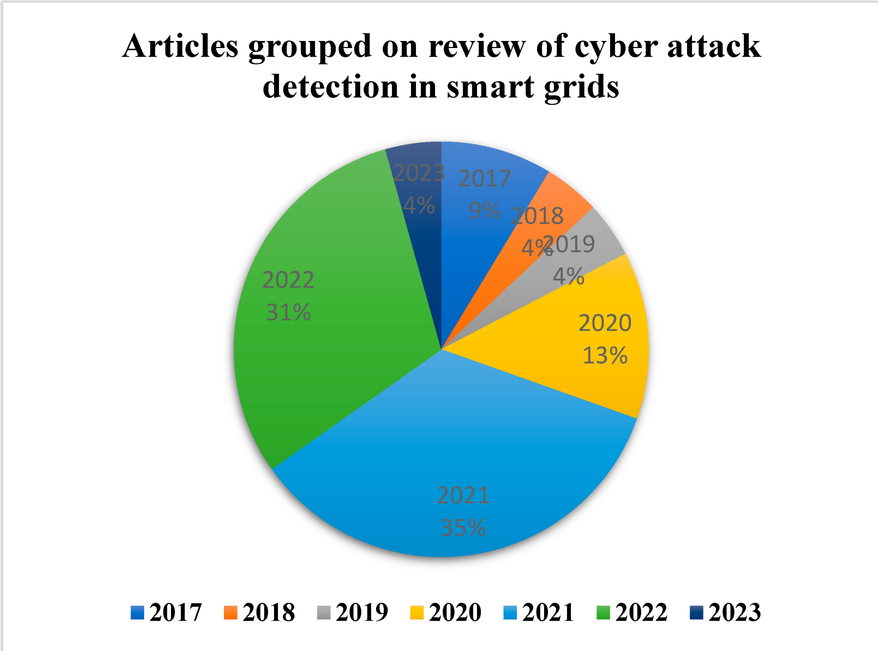

Figure 1.4: Estimate: cyber attacks will increase exponentially [4]

choice, are measures necessary to ensure that competing electricity providers have equal access to customer data compared to the incumbent utility?

In competitive environments among electricity providers, access to users' electricity usage patterns and behavioral information holds significant importance. Providers or their representatives may use this data to develop business strategies and create tailored packages or offers. In an open market scenario, some data may be disclosed after offers are made public, providing a level playing field for information access. However, if privacy is compromised beforehand, with specific user data available to only certain parties, these electricity providers could potentially gain unfair advantages. Therefore, effective privacy policies are essential to prevent the exploitation of unfair means in shaping business strategies.

The integration of Information and Communication Technologies (ICTs) into smart grid operations introduces various privacy concerns. Depending on how a consumer uses and recharges electricity [17].

1.7.5 Cost of Implementation:

Estimating Smart Grid costs poses challenges due to several factors. Integrating digital technology into Smart Grids introduces complexities, as the failure rates and life expectancy of embedded assets differ from traditional grid technologies. For instance, a substation transformer designed for 40 years may be coupled with information technology lasting 10, 15, or 20 years, necessitating careful cost considerations for upgrades. Additionally, the rapid obsolescence of digital tech complicates estimates, as advancing communications and computational capabilities may render Smart Grid components obsolete before their intended lifespan ends.

Moreover, the evolution of Smart Grid technologies is expected to outpace conventional tech in terms of cost reduction and advancements. However, uncertainties persist, particularly with new and unproven Smart Grid technologies. If their performance is subpar or degrades unexpectedly, it could jeopardize the entire technology's viability and business plan. As Smart Grid component costs decrease rapidly due to maturation and increased production, estimating replacement costs becomes challenging[5].

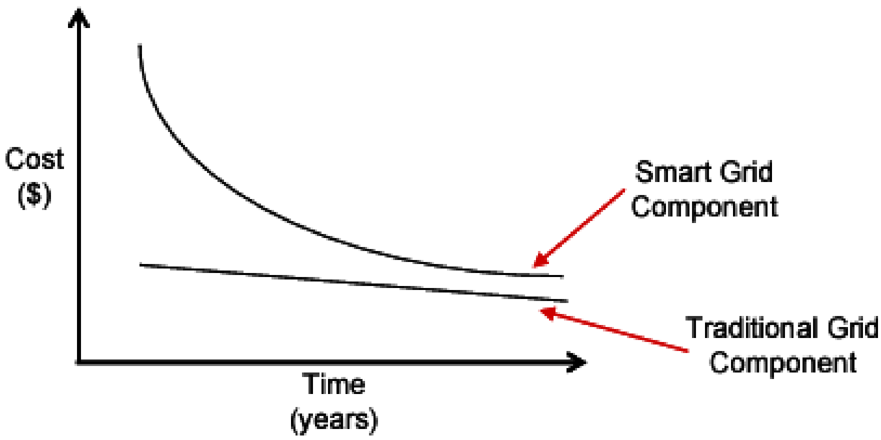

Figure 1.5: Grid Component Costs [5]

1.7.6 Regulatory Frameworks

Electric distribution systems across Europe are encountering significant hurdles stemming from climate change objectives, evolving market frameworks, and technological advancements. These factors will have a profound impact on the responsibilities of distribution system operators. The challenges' nature and magnitude are primarily influenced by Europe's vision and strategies regarding climate and energy. This research aims to identify which policies might pose obstacles to innovation in distribution grids and the adoption of advanced smart grid solutions developed within the UNITED-GRID project. Following an in-depth examination of emerging policy priorities within the energy and climate framework, as well as electricity market design, and subsequent consultations with three partner distribution system operators, five key barriers have been pinpointed. The findings indicate that ambitious decarbonisation targets and shifting expectations regarding the role of distribution system operators in the energy landscape necessitate more adaptable and efficient network management. However, rigid income frameworks, insufficient incentives for innovation, and regulatory uncertainties impede the modernisation of distribution systems. It can be inferred that these concerns heighten the risks for distribution system operators and must be taken into account by research initiatives and developers of smart grid solutions to successfully implement and achieve market adoption of the developed solutions [18].

Conclusion

The smart grid revolution is not a destination, but a continuous journey towards a more efficient, reliable, and sustainable energy future. While challenges exist, the potential benefits of the smart grid are undeniable. By embracing innovation, fostering collaboration, and empowering consumers, we can unlock the full potential of this transformative technology.

A smarter grid paves the way for a future where clean energy sources like solar and wind power are seamlessly integrated, homes and businesses actively participate in energy management, and power outages become a rarity. It's a future where we have a more secure and sustainable energy infrastructure for generations to come.

Let's continue exploring the exciting world of smart grids and work together to build a brighter energy future for all.

Chapter

2

Implementation

Introdution

In this chapter, we present our contributions towards the deep learning-based intrusion detection system for the smart grid. due to the fact that smart grid communication requires connecting to the internet, smart grid components now have a new weakness which is cyber-threat. In order to counter this issue, we suggest a deep learning method that utilizes Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) which is a specific type of Recurrent Neural Network (RNN) to enhance the precision and efficiency of identifying intrusions within the smart grid communication infrastructure.

2.1 Theoretical Proposal

2.1.1 Project Description

the proposed system for the project is a network intrusion detection system based on deep learning that will depend on either CNN or LSTM as a learning algorithme. This project mainly aims at designing and implementing said system that can detect cyber threats in Smart Grid Communication Infrastructure effectively.To capture both spatial and temporal features of network traffic data, the proposed system will use either CNN or LSTM architectures to identify complex and sophisticated attack patterns that would threaten the smart grid functionality. Our deep learning based network intrusion detection system will be mainly focused on denial of service attacks (DoS) and distributed denial of service attacks (DDoS).

2.1.2 Project Design and architecture

Building any machine learning or deep learning model usually involves several important steps, as shown in Figure 2.1.

First, we need to collect data that is relevant to the function of the model we want to train, which we will then need to preprocess, which entails clearning, encoding, augmenting, and standardising the data to prepare it for the training phase.

The next step is to select a learning algorithm, the proper optimizer, a loss function, and the evaluation metrics that we will use to train our model.

The model is then trained on the cleaned data, and this is where we feed the trained model new data it has not yet seen before and get the results and predictions on the new data.

The next and final step is the validation step, in which we get the prediction results and evaluate the model accuracy, recall, and f1-score.

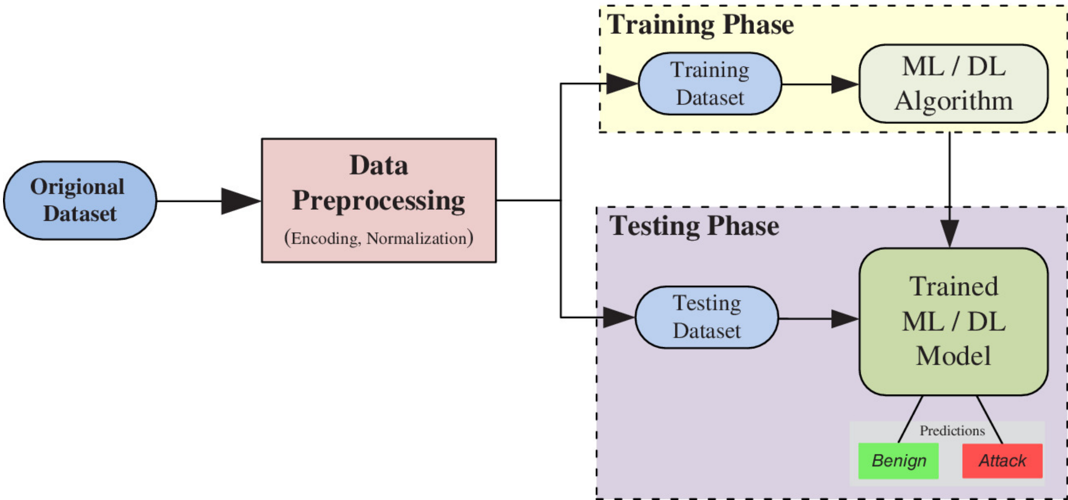

Figure 2.1: deep learning model creation

2.1.3 Deep learning Models architecture

Detecting cyber threats to the smart grid functionality and safety is crucial a task which needs high accuryacy of detection that's why we opted to we choose two deep learning algorithms for intrusion detection which are CNN and LSTM

2.1.3.1 CNN model

Convolutional Neural Networks (CNNs) are a highly-specific deep learning algorithm meant to analyze spatialy structured input data, like images and grid structured data. These are derived from the visual cortex of human beings and are good at image recognition, object detection and image segmentation among others. CNNs work by carrying out convolutional layers in order to capture local characteristics, pooling layers in order to reduce spatial dimensions and fully connected layers for predictions. They can be trained on labeled data and have proved very effective in several applications like face recognition, medical imaging analysis as well as self-driving cars.[19]

Note: will expande on this later

2.1.3.2 LSTM model

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that was developed by Hochreiter and Schmidhuber; LSTM is different from RNNs because it can forecast sequences and study long-term dependency patterns from the provided data. What makes LSTMs unique is their ability to learn order dependency patterns which is critical in solving complex problems like speech recognition and machine translation. LSTM address the weakness of traditional RNN which is the unability to learn any long term patterns which it solves by introducing a memory cell, LSTM is controlled with Three gates control: input gate, forget gate, output gate.These gates determine what information to add, remove, and output from the memory cell and hence enable LSTM networks to learn long-term dependency patterns. [20]

Note: will expande on this later

2.2 Implementation and Experiments

2.2.1 Development tools used

2.2.1.1 Development environment

training deep learning models requires a high performance PC due to the fact that these models require high computational resources to process large datasets and intricate neural networks. With complex calculations and huge amounts of data. One of the reasons why having a powerful PC with a powerful GPU is important during the training process is that it can significantly shorten the time needed for the training, as well as reduce computational resources. a sufficient quantity of memory (RAM) is also needed to load big datasets, and fast storage devices like SSDs are vital in handling data-intensive nature of deep learning tasks.

- jupyter notebook: is an open source web application or a vscode extension that facilitates the creation and sharing of segmented documents that contains blocks of interactive code, text and data visualations, it is mostly used for data science, machine learning and scientific computing, it supports a wide range of programming languages like python, R, scala and julia, it can also display some text formats like markdown, Latex and HTML.

- VScode: Visual Studio Code (VS Code) is an open-source source-code editor developed by Microsoft for Windows, Linux, macOS, and web browsers. It is a popular choice among developers due to its extensive features like code highlighting, debugging, code completion, and the ability to extend its original functionality with 3rd-party extensions and extensibility. It also offers Git integration out of the box.

2.2.1.2 programming languages

The programming language we mainly used is Python 3, important aspect of Python lies in its being more than just an object-oriented programming language because it supports other programming paradigms, which are procedural and functional programming making it flexible when choosing the desired approach. It has user-friendly syntax making it easy to digest even for beginner, This has led to the rise of Python's ecosystem fostered by active community coupled with simplicity behind coding style making it easily accessible by almost everyone. To further improve its capability and functionality, python boast a wide range of third party packages that can easily be installed through Python package manager called pip. Python was created in 1991 by Guido van Rossum. A major landmark came in 2008 with the development of python3 which introduced several improvements and enhancements to the language thereby cementing its place as a valuable flexible programming tool on earth today. [21]

libraries that are used for the development:

- NumPy: NumPy is the primary array programming library for the Python language, with an essential role in research analysis pipelines across diverse fields such as physics, chemistry, astronomy, geoscience, biology, psychology, and more. The NumPy array is an efficient data structure that stores and accesses multidimensional arrays (tensors), enabling a wide variety of scientific computation. NumPy was initially developed by students, faculty and researchers to provide an advanced, open-source array programming library for Python, with a sense of building something consequential together for the benefit of many others. [21]
- Pandas: Pandas is a Python open source program meant for data management and analysis, it was started in 2008 by AQR Capital Management. It went public in late 2009 and has an active community of contributors. Some of the most important features associated with pandas are fast and efficient DataFrame object for data manipulation with integrated indexing, tools for reading and writing data between in-memory data structures and different formats, time series functionality like date

range generation, frequency conversion, moving window statistics, and date shifting and highly optimized performance with critical code paths written in Cython or C. [22]

- Matplotlib: Matplotlib is a 2D plotting library for Python which can produce publication quality plots, used in application development, interactive scripting and image creation on all operating system and user interface platforms. The author of Matplotlib John D. Hunter began using Python in 2001 and was initially frustrated at the lack of a powerful graphics environment like MATLAB's. He then developed Matplotlib to satisfy his needs, focusing initially on embedding it in a GUI for his ECoG application and then gradually adding support for other features like high-quality raster and vector output, support for mathematical expressions, and interactive use from the shell.[23]
- Seaborn: Seaborn is a python library for making statistical graphics, Seaborn is a high-level interface to Matplotlib and compatible with Pandas's data structures. Seaborn can provide the data with a dataset alongside plot specifications and automatically maps the values onto visual attributes such as color, size or style; internally seaborn performs the relevant statistical transformations and finally adds axis labels as well as legends for the plots. Many Seaborn functions can generate multi-panel figures for comparing different subsets of data or variable pairings within a dataset. By allowing quick prototyping and exploratory analysis of data in single-function calls with just a few arguments, Seaborn can be used throughout the scientific project cycle. [24]
- scikit-learn: Sckit-learn is a Python library that provides various machine learning algorithms for medium-scale supervised and unsupervised problems. It focuses on making things easy, having good performance, documentation and remaining consistent in it's APIs. Scikit-learn depends on scientific Python ecosystem libraries such as NumPy and SciPy, and uses Cython to blend C/C++ with Python for improved performance. This lightweight software has few requirements and can be obtained by anyone without any major legal challenges since it is distributed under a simplified BSD license. It provides solid implementations of Machine Learning algorithms, documentation and community driven development, scikit-learn also includes some nice implementations of different algorithms outperforming other popular python ML libraries in many instances including SVMs, Lasso, Elastic Net, k-Nearest Neighbors, PCA, k-Means and some other algorithms.[25]
- TensorFlow: TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It gives you more flexibility and control the some of the other machine learning frameworks with features such as the Keras Func-

tional API and the Sub-Classification API model to build complex neural network topologies. it offers fast execution for fast debugging and simple prototyping.[26]

- Keras: Keras is an open-source library that provides a Python interface for artificial neural networks. Keras was first independent software, then it was integrated into TensorFlow library, and later started supporting others like AJX and PyTorch.[27]

2.2.2 Dataset

The IDS 2018 dataset is the data set used for this project, this dataset is a comprehensive and realistic dataset for intrusion detection systems. it was created through a collaboration effort between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). It includes a few types of attacks such as Brute-force, Botnet, DoS, DDoS and Web attacks, also network infiltration from within all of which are a common attack on smart grid systems. This resulted in 16,233,002 traffic samples which were collected over 10 days from ten real networks, an unusual feature of this data set is its imbalance in benign to malicious ratio of cases. The CICFlowMeter-V3 generates 80 features extracted from the network traffic which describe various intrusions along with abstract distribution models for applications, protocols or even lower level networking entities. Researchers widely employ this dataset to analyze their IDS performance in different research works while others use it to build advanced IDS models. This dataset is not specific to smart grid activity but it is a generalized dataset that includes generalized network traffic which would be the same in a smart grid.

2.2.3 Data preprocessing

Data preprocessing is a crucial step, and the first step in training a machine learning model is data preprocessing. It involves cleaning, transforming, and organising the dataset before it can be utilised by the machine learning algorithms. Data preprocessing entails improving dataset quality by addressing issues such as missing values, invalid values, and inconsistencies. Data preprocessing techniques include cleaning the data to get rid of errors, normalising the data so that features have the same scale, and feature engineering, which will result in new informative variables, augmenting the data, and resampling it to avoid bias in our model. Preparing the data effectively ensures that machine learning models can accurately learn patterns, increasing their performance and, hence, more accurate results.

2.2.3.1 importing data

First we load the data with Pandas library, and since our dataset is split into 10 files, we load the files which include the data related to DoS and DDoS attacks and we merge the data into the same variable for easier preprocessing while deleting the old variable to evoid uselessly fillig the memory, we also remove unneeded column from one of the dataset files

```
network_data1 = pd.read_csv('02-15-2018.csv', low_memory=False)
network_data2 = pd.read_csv('02-16-2018.csv', low_memory=False)
network_data3 = pd.read_csv('02-20-2018.csv', low_memory=False)
network_data4 = pd.read_csv('02-21-2018.csv', low_memory=False)

network_data3.drop(columns=['Flow ID', 'Src IP', 'Src Port', 'Dst IP'],
    axis=1,inplace=True)

network_data = pd.concat([network_data1, network_data2], axis=0)
network_data.reset_index(drop=True, inplace=True)
del network_data1, network_data2

network_data = pd.concat([network_data, network_data3], axis=0)
network_data.reset_index(drop=True, inplace=True)
del network_data3

network_data = pd.concat([network_data, network_data4], axis=0)
network_data.reset_index(drop=True, inplace=True)
del network_data4
```

the total amount of data loaded is about 11 million rows with 80 columns all of which is either benign or DoS/DDoS traffic with a total size of 6.6 GB.

	Dst Port	Protocol	Timestamp	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	0	0	15/02/2018 08:25:18	112641158	3	0	0.0	0.0	0.0	0.0	...	0	0.0	0.000000	0.0	0.0	56320579.0	7.042784e+02	56321077.0	56320081.0	Benign
1	22	6	15/02/2018 08:29:05	37366762	14	12	2168.0	2993.0	712.0	0.0	...	32	1024353.0	649038.754495	1601183.0	321569.0	11431221.0	3.644991e+06	15617415.0	8960247.0	Benign
2	47514	6	15/02/2018 08:29:42	543	2	0	64.0	0.0	64.0	0.0	...	32	0.0	0.000000	0.0	0.0	0.0	0.000000e+00	0.0	0.0	Benign
3	0	0	15/02/2018 08:28:07	112640703	3	0	0.0	0.0	0.0	0.0	...	0	0.0	0.000000	0.0	0.0	56320351.5	3.669884e+02	56320611.0	56320092.0	Benign

Figure 2.2: data sample

as we can see in the Figure 2.3 and Table 2.1 below, the data is unbalnced, but we will fix that later in the data preprocessing phase, specifically in the data augmentation phase.

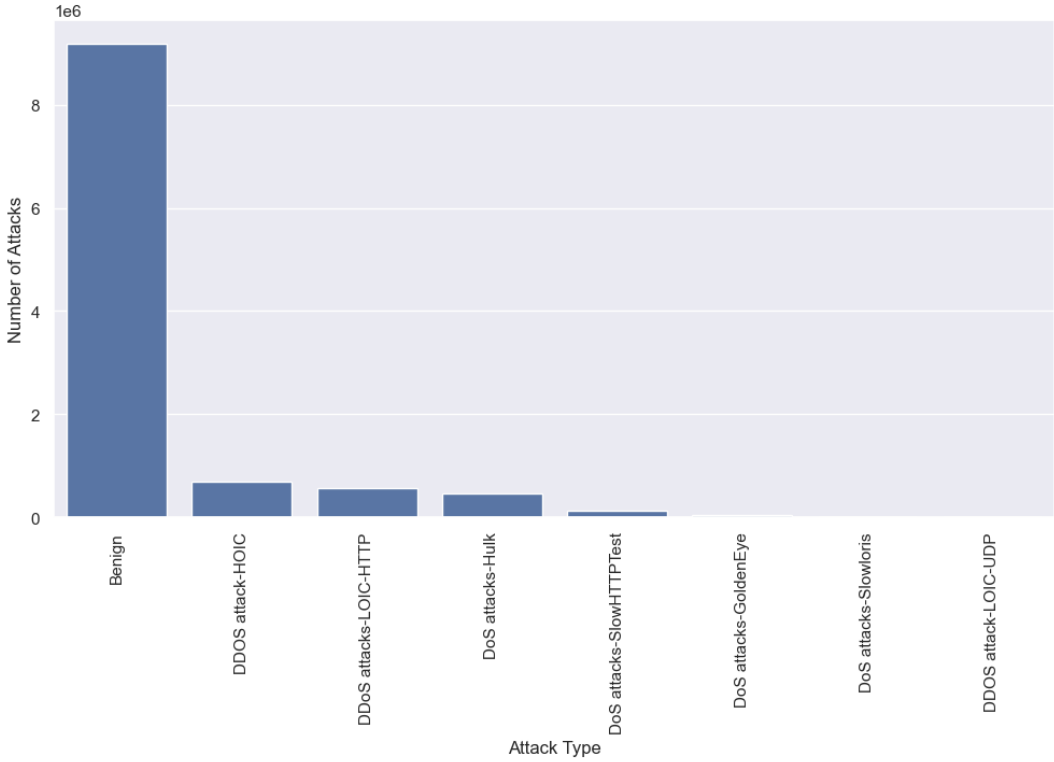

Figure 2.3: data sample

Table 2.1: the number of occurrence for each traffic type

activity type	number of ocurence
Benign	9176239
DDOS attack-HOIC	686012
DDoS attacks-LOIC-HTTP	576191
DoS attacks-Hulk	461912
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDOS attack-LOIC-UDP	1730

2.2.3.2 Cleaning data

To clean our data we must first find and remove unwanted data like missing values, null values, duplicate rows and unneeded columns or features.

- finding and cleaning missing values: First we identify the columns that contain null values, in a our dataset by identifying columns with missing values,after which we decide to eliminat the rows containing null values, in a Dataset. The objective of this step is to eliminate missing data to ensure quality and consistency Data for the model training.

```
# find null or missing values
network_data.isna().sum().to_numpy()

# drop null or missing columns
cleaned_data = network_data.dropna(inplace=true)
```

- removing duplicate rows: we also remove duplicate rows for a better quality dataset and to avoid bias in our model.

```
# removing duplicate rows
cleaned_data.drop_duplicates(inplace=True)
```

2.2.3.3 Encoding the categorical variables

The LabelEncoder assigns a unique integer to each categorical value which is the label in our case which is the traffic type(benign or attack), which which allows them to be represented in a numerical form. This makes it easier to use of these variables in machine learning algorithms, as they can handle numerical values better.

```
le = LabelEncoder()
cleaned_data['Label']= le.fit_transform(cleaned_data['Label'])
cleaned_data['Label'].unique()
```

2.2.3.4 Augmenting the data

As we have stated before in section 2.2.2 that our dataset is unbalanced in the distribution between benign and malicious activity which is a bad thing because it will introduce bias, overfitting and poor prediction performance to our model.

In this step we will resample our dataset to get a better 1:1 ration between our different categorical variables (benign and other attack types).

```
from sklearn.utils import resample

data_1_resample = resample(data_1, n_samples=20000, random_state=123,
    replace=True)
data_2_resample = resample(data_2, n_samples=20000, random_state=123,
    replace=True)
data_3_resample = resample(data_3, n_samples=20000, random_state=123,
    replace=True)
data_4_resample = resample(data_4, n_samples=20000, random_state=123,
    replace=True)
data_5_resample = resample(data_5, n_samples=20000, random_state=123,
    replace=True)
data_6_resample = resample(data_6, n_samples=20000, random_state=123,
    replace=True)
```

```
data_7_resample = resample(data_7, n_samples=20000, random_state=123,
    replace=True)
data_8_resample = resample(data_8, n_samples=20000, random_state=123,
    replace=True)
```

the python code snippet uses the resample function from scikit-learn, it does the resampling we need to make our dataset balanced

- data x: all of those variables are our data which has been cleaned and separated according to the previously encoded categorical variables.
- n samples: is the number of rows for each attack types, in this case we used 20000 rows.
- random state: this is the resampling seed, by using the same seed number we can ensure that we always get the same results.
- replace: replace decides whether or not the samples can be selected multiple times.

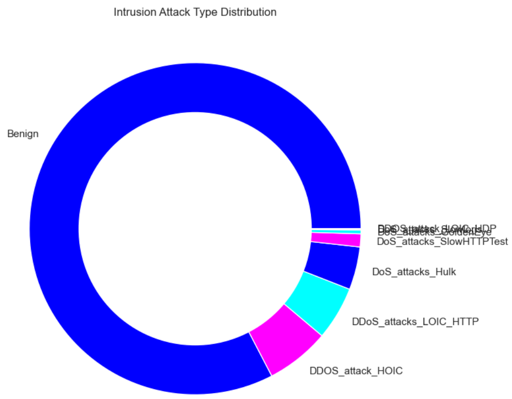

Figure 2.4: Before data augmentation

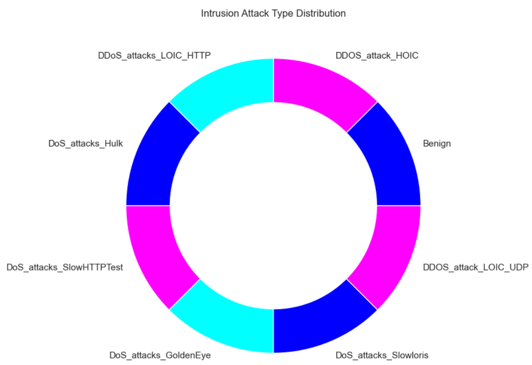

Figure 2.5: After data augmentation

those resampled variables are then merged and are fed to the deep learning model for training.

2.2.3.5 splitting the data for the deep learning model

In this step we split our data into 2 sets:

- training data: 90% of the total dataset
- testing data 10% of the total dataset

```
test_dataset = train_dataset.sample(frac=0.1)
target_train = train_dataset['Label']
target_test = test_dataset['Label']
target_train.unique(), target_test.unique()
```

2.2.4 Deep learning models implementation

2.2.4.1 CNN model

We built a deep learning model that uses CNN as a learning algorithm with 64 filters, ReLU as the activation function in Python

CNN model summary

```
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=6, activation='relu',
    padding='same', input_shape=(72, 1)))
model.add(BatchNormalization())

model.add(MaxPooling1D(pool_size=(3), strides=2, padding='same'))

model.add(Conv1D(filters=64, kernel_size=6, activation='relu',
    padding='same', input_shape=(72, 1)))
model.add(BatchNormalization())
model.add(MaxPooling1D(pool_size=(3), strides=2, padding='same'))

model.add(Conv1D(filters=64, kernel_size=6, activation='relu',
    padding='same', input_shape=(72, 1)))
model.add(BatchNormalization())
model.add(MaxPooling1D(pool_size=(3), strides=2, padding='same'))

model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(8, activation='softmax'))
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam',
    metrics=['accuracy'])
```

deep learning model creation:

- Sequential(): Creates a Sequential model which is a linear stack of layers.
- Conv1D(): adds one dimensional convolutional layer to the model, sets the feature filter and the activation function
- MaxPooling1D(): adds one dimensional pooling layer to the model, which is used to reduce the spatial dimensions of the data
- Flatten(): used to convert the multidimensional output of the previous layers into a 1D vector
- Dense(): a fully connected layer used to do the classification tasks
- BatchNormalization(): used to normalize the data between different layers, it helps improve the model's performance
- compile(): used to configure the model, it can set the following parameters:
  - loss function: categorical crossentropy is used to compute the cross-entropy loss between the true class distribution and the predicted class distribution
  - optimizer: adam adapts the learning rate for each parameter individually making it effective for fast convergence and robustness.
  - metrics: accuracy is the precision of the predicion as a ratio of correct prediction to the total number of predictions which is a measure of the overall performance of the model.

we also get a summary of the created model, this summary describes the arrangement of the model layers, the number of parameters of each layer, the output shape of each layer and the number of trainable and non-trainable parameters

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 72, 64)	448
batch_normalization (BatchNormalization)	(None, 72, 64)	256
max_pooling1d (MaxPooling1D)	(None, 36, 64)	0
conv1d_1 (Conv1D)	(None, 36, 64)	24,640
batch_normalization_1 (BatchNormalization)	(None, 36, 64)	256
max_pooling1d_1 (MaxPooling1D)	(None, 18, 64)	0
conv1d_2 (Conv1D)	(None, 18, 64)	24,640
batch_normalization_2 (BatchNormalization)	(None, 18, 64)	256
max_pooling1d_2 (MaxPooling1D)	(None, 9, 64)	0
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 64)	36,928
dense_1 (Dense)	(None, 64)	4,160
dense_2 (Dense)	(None, 8)	520

Total params: 92,104 (359.78 KB)

Trainable params: 91,720 (358.28 KB)

Non-trainable params: 384 (1.50 KB)

Figure 2.6: CNN model summary

Next step is starting the model training with 30 epochs, 32 batch sier and the validation data which is the test data we split from the original dataset earlier

```
his = model.fit(X_train, y_train, epochs=30, batch_size=32,
    validation_data=(X_test, y_test))
```

2.2.4.2 LSTM model

The LSTM mode implementations is very similar to the CNN implementation with only a few challenges, with those changes being that CNN uses Conv1D() to create it's one dimensional convolutional layer, while LSTM uses LSTM() function to add it's layers

```
model = Sequential()
model.add(LSTM(units=64, return_sequences=True, input_shape=(72, 1)))
model.add(BatchNormalization())

model.add(LSTM(units=64, return_sequences=True))
model.add(BatchNormalization())
```

```
model.add(LSTM(units=64))
model.add(BatchNormalization())

model.add(Dense(64, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(8, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam',
    metrics=['accuracy'])
```

we get a model summary for the LSTM model as well

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 72, 64)	16,896
batch_normalization_3 (BatchNormalization)	(None, 72, 64)	256
lstm_4 (LSTM)	(None, 72, 64)	33,024
batch_normalization_4 (BatchNormalization)	(None, 72, 64)	256
lstm_5 (LSTM)	(None, 64)	33,024
batch_normalization_5 (BatchNormalization)	(None, 64)	256
dense_3 (Dense)	(None, 64)	4,160
dense_4 (Dense)	(None, 64)	4,160
dense_5 (Dense)	(None, 8)	520

Total params: 92,552 (361.53 KB)

Trainable params: 92,168 (360.03 KB)

Non-trainable params: 384 (1.50 KB)

Figure 2.7: LSTM model summary

2.2.5 Results

In this we will compile the data from our models evaluation on the test data with both the CNN and the LSTM models, the metrics that we use for evaluation are mainly the detection accuracy and loss rate, we will also be looking at other mertics like some criteria of evaluation are:

- Accuracy: Accuracy measures the overall correctness of a model's predictions. It is calculated as (TP + TN) / (TP + TN + FP + FN). Accuracy can be misleading, especially with imbalanced datasets.
- precision: Precision measures the proportion of true positives among the positive predictions made by the model. It is calculated as TP / (TP + FP). Precision is useful when the cost of false positives is high, such as in spam detection.
- F1-score: The F1 score is the harmonic mean of precision and recall. It ranges from 0 to 1, with 1 being best. The F1 score provides a balance between precision and recall. It is calculated as 2 * (Precision * Recall) / (Precision + Recall).

2.2.5.1 CNN model

evaluating the CNN model with the test data:

- accuracy: 99.66%
- loss: 1.44%

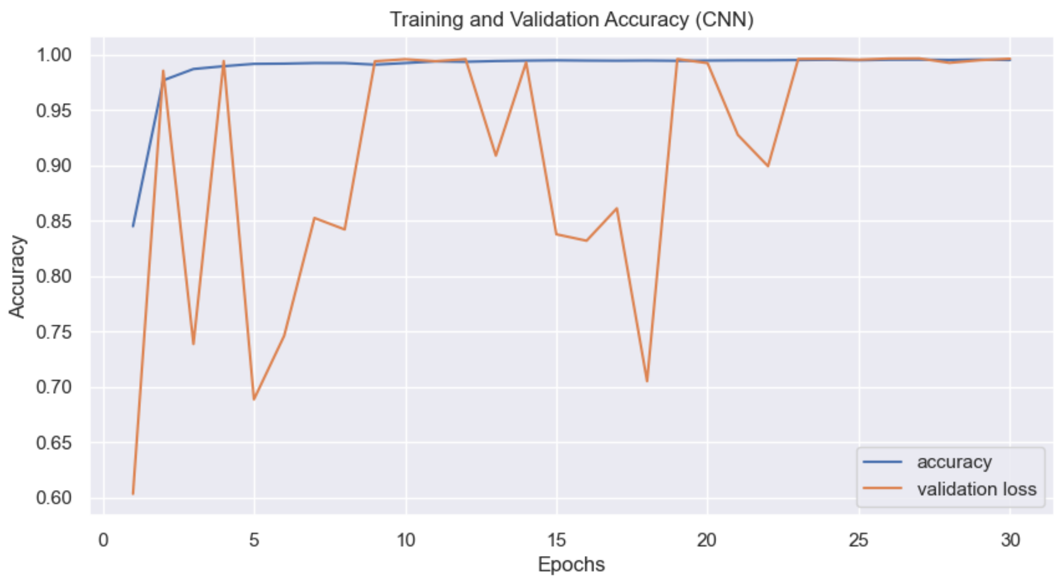

Figure 2.8: Accuracy curve

Figure 2.9: Loss curve

2.2.5.2 LSTM model

evaluating the LSTM model with the test data:

- accuracy: 99.62%
- loss: 1.6%

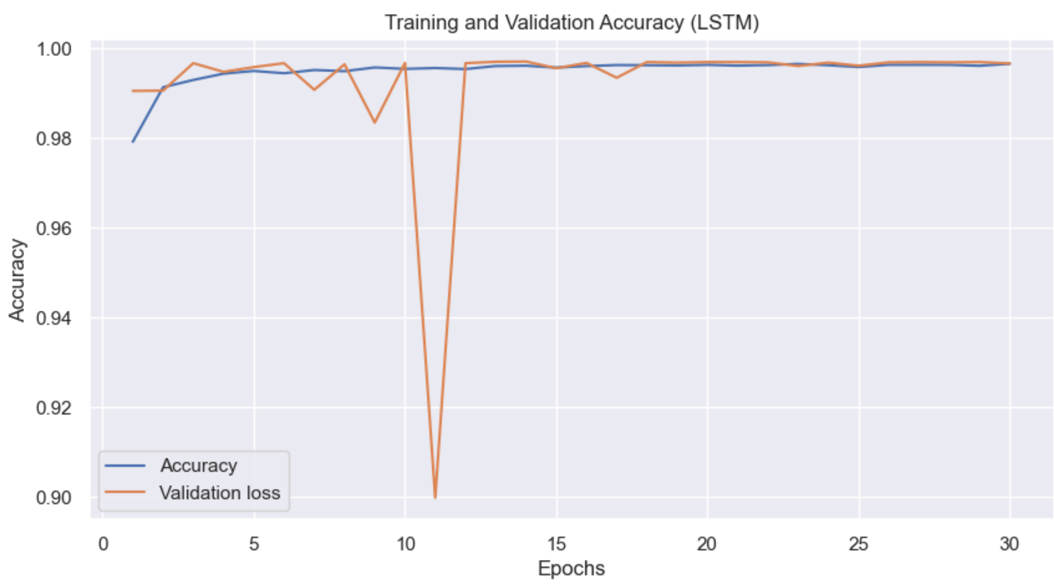

Figure 2.10: Accuracy curve

Observation : We notice that the accuracy and the loss rates are almost matched between the CNN and the LSTM models. However according the validation accuracy compared to

Figure 2.11: Loss curve

the number of epochs, the LSTM provides better accuracy over a wide rande of epochs(10 to 12 and 25 to30) unlike CNN which provides it's best accuracy over a narrow range of epochs(3 to 7 and 12 to 30).

2.2.6 Conclusion

In this chapter we demonstrated the development of two different deep learning methodes for creating a network intrusion detection system that protects the smart grid from DoS and DDoS attacks, starting with the development environment, the architecture of the used algorithms, and the data preprocessing, cleaning and training the models, and finishing the chapter with a performance comparison between the two models.

General Conclusion

The proposed deep learning-based network intrusion detection system effectively addresses the issue of protecting smart grid infrastructure from Distributed Denial-of-Service (DDoS) attacks. The system leverages Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) algorithms to identify and prevent malicious traffic. This solution is particularly relevant in the context of smart grids, where the reliability and security of communication networks are crucial for efficient and safe operation.

contribution

- Development of a deep learning-based intrusion detection system: The system uses a combination of CNN and LSTM algorithms to identify DDoS attacks in smart grid communication networks.
- Improved accuracy and efficiency: The proposed system demonstrates enhanced accuracy and efficiency compared to traditional methods, making it a more effective solution for detecting and preventing DDoS attacks.
- Scalability and adaptability: The system is designed to be scalable and adaptable to various smart grid communication network configurations and traffic patterns.

Perspectives

The applicability of this project is significant, as it can be integrated into existing smart grid infrastructure to enhance the security and reliability of communication networks. This can lead to improved overall performance and reduced downtime, ultimately benefiting both the grid operators and consumers. However, there are some limitations and perspectives to consider:

Limitations

- Data quality and availability: The quality and availability of training data can significantly impact the performance of the system. Future work should focus on developing methods to handle noisy or limited data.
- Real-time processing: The system's ability to process data in real-time is crucial for effective DDoS attack detection. Future improvements should focus on optimizing the system's processing speed and efficiency.
- Integration with existing systems: The system should be designed to seamlessly integrate with existing smart grid infrastructure and security systems to ensure a smooth transition and optimal performance.
- Future research directions: Future research should explore the application of other deep learning architectures and techniques to further improve the system's performance and adaptability.

By addressing these limitations and perspectives, the proposed deep learning-based network intrusion detection system can be further refined and optimized to provide enhanced security and reliability for smart grid communication networks.

Bibliography

[1] Avi Gopstein, Cuong Nguyen, Cheyney O'Fallon, Nelson Hastings, David Wollman, et al. *NIST framework and roadmap for smart grid interoperability standards, release 4.0*. Department of Commerce. National Institute of Standards and Technology . . . , 2021.

[2] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid—the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4): 944–980, 2011.

[3] Zhiyi Chen, Ali Moradi Amani, Xinghuo Yu, and Mahdi Jalili. Control and optimisation of power grids using smart meter data: A review. *Sensors*, 23(4):2118, 2023.

[4] Link to mdpi article. URL `https://www.mdpi.com/1996-1073/16/4/1651`. Accessed on April 20, 2024.

[5] U.S. Department of Energy. Estimating the costs and benefits of the smart grid: A preliminary estimate, 2011. URL `https://smartgrid.gov/files/documents/Estimating_Costs_Benefits_Smart_Grid_Preliminary_Estimate_In_201103.pdf`. Accessed on 20 Avril 2024.

[6] Joe Miller. Understanding the smart grid: Features, benefits and costs. In *Illinois Smart Grid Initiative–Workshop*, 2008.

[7] Hamid Gharavi and Reza Ghafurian. *Smart grid: The electric energy system of the future*, volume 99. IEEE Piscataway, NJ, USA, 2011.

[8] Mohamed E El-Hawary. The smart grid—state-of-the-art and future trends. *Electric Power Components and Systems*, 42(3-4):239–250, 2014.

[9] Haotian Zhang. *Smart Grid Technologies and Implementations*. PhD thesis, City University London, 2014.

[10] GM Shafiullah, Aman Maung Than Oo, ABMS Ali, and Peter Wolfs. Smart grid for a sustainable future. 2013.

[11] Blackridge Research. What is a smart grid? what are the major smart grid technologies? URL `https://www.blackridgeresearch.com/blog/what-is-a-smart-grid-what-are-the-major-smart-grid-technologies#smart-grid-technologies`. Accessed on 22 April 2024.

[12] Artur R Avazov and Liubov A Sobinova. Advanced distribution management system. In *EPJ Web of Conferences*, volume 110, page 01004. EDP Sciences, 2016.

[13] ElProCus. Overview of smart grid technology, operation & application in existing power system. URL `https://www.elprocus.com/overview-smart-grid-technology-operation-application-existing-power-system/`. Accessed on 20 April 2024.

[14] Lena Wang. Smart grids and circuit breakers: The tale of two happy enemies. URL `https://www.linkedin.com/pulse/smart-grids-circuit-breakers-tale-two-happy-enemies-lena-wang-iuy1c?trk=article-ssr-frontend-pulse_more-articles_related-content-card`. Accessed on 20 April 2024.

[15] Yu Cunjiang, Zhang Huaxun, and Zhao Lei. Architecture design for smart grid. *Energy Procedia*, 17:1524–1528, 2012.

[16] Kenneth Kimani, Vitalice Oduol, and Kibet Langat. Cyber security challenges for iot-based smart grid networks. *International journal of critical infrastructure protection*, 25:36–49, 2019.

[17] Sherali Zeadally, Al-Sakib Khan Pathan, Cristina Alcaraz, and Mohamad Badra. Towards privacy protection in smart grid. *Wireless personal communications*, 73:23–50, 2013.

[18] Joni Rossi, Ankur Srivastava, David Steen, and Le A Tuan. Study of the european regulatory framework for smart grid solutions in future distribution systems. In *CIRED 2020 Berlin Workshop (CIRED 2020)*, volume 2020, pages 800–802. IET, 2020.

[19] Osval Montesinos-López, Abelardo Montesinos, and Jose Crossa. *Convolutional Neural Networks*, pages 533–577. 01 2022. ISBN 978-3-030-89009-4. doi: 10.1007/978-3-030-89010-0_13.

[20] Deep learning | introduction to long short term memory. URL `https://www.geeksforgeeks.org/`

deep-learning-introduction-to-long-short-term-memory/. accessed: 01-06-2024.

[21] General python faq. URL https://docs.python.org/3/faq/general.html#what-is-python. accessed: 28-05-2024.

[22] About pandas. URL https://pandas.pydata.org/about/. accessed: 28-05-2024.

[23] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[24] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] Why tensorflow. URL https://www.tensorflow.org/about. accessed: 01-06-2024.

[27] Introducing keras 3.0. URL https://keras.io/keras_3/. accessed: 01-06-2024.