Learning-Based Control of Microgrids with Transformers and MPC

B,W.C. van der Woude

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Learning-Based Control of Microgrids with Transformers and MPC

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

B,W.C. van der Woude

June 27, 2024

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

In the evolving landscape of energy systems, microgrids have emerged as a key solution for enhancing energy efficiency and sustainability. Capable of operating independently or alongside the main power grid, microgrids integrate renewable energy sources and ensure local energy distribution. This makes them instrumental in reducing dependencies on centralised power supplies and improving resilience against disruptions. This research addresses the unit commitment problem, a mathematical optimisation challenge where the objective is to coordinate a group of energy production units to meet demand at minimal cost. We model the microgrid as a mixed logical dynamical (MLD) system, incorporating both the continuous and discrete variables involved in the microgrid. model predictive control (MPC) is selected as the control strategy due to its suitability for controlling hybrid systems and its ability to handle complex constraints.

However, the application of MPC is challenged by the need to solve computationally demanding mixed-integer linear programming (MILP) problems at each control iteration, which are combinatorial. To address this challenge, this research proposes integrating a learning-based method to enhance MPC in microgrids. We propose using transformers to learn and predict the binary decisions in MILP problems, thereby reducing the problem to a more tractable linear programming (LP) problem. Transformers are chosen for their ability to recognise patterns in sequential data, a key aspect of the decision-making process in MPC. Furthermore, their capability for parallel processing allows for more efficient training and scalability to larger problems, making them highly suitable for handling the dynamic and complex optimisation tasks found in microgrid control. Simulation experiments show that integrating transformers in the decision of the discrete variables reduces the overall computation with only a slight loss of optimality and, therefore, improves the online applicability of MPC in microgrid control.

Table of Contents

1 Introduction **1**
1-1 Background . . . 1
1-2 Problem Description . . . 2
1-3 Thesis Outline . . . 3

2 Background and Literature Review **5**
2-1 System Model: Microgrids . . . 5
2-1-1 System Description . . . 6
2-1-2 Control of Microgrids using MPC . . . 8
2-2 Related Work . . . 9
2-2-1 Mixed-Integer Linear Programming . . . 10
2-2-2 Learning-based MPC . . . 10
2-2-3 Conclusions from the Literature . . . 13
2-3 Transformers . . . 13
2-4 Summary . . . 17

3 Methodology: Transformers for Binary Sequence Prediction **19**
3-1 General Overview Methodology . . . 19
3-2 Data Collection and Preprocessing . . . 20
3-2-1 Data Collection . . . 21
3-2-2 Data Preprocessing . . . 21
3-3 Transformer Architecture for MILP in Microgrids . . . 26
3-4 Training and Prediction of the transformer . . . 27
3-4-1 Transformer Training . . . 27
3-4-2 Transformer Prediction . . . 29
3-5 Integration with MPC . . . 30
3-6 Evaluation Metrics and Validation . . . 32
3-7 Conclusion . . . 33

4 Results and Simulations: Transformer-Based MPC for Microgrids 35
4-1 Overview of Experimental Setup . . . 35
4-1-1 Simulation Environment . . . 36
4-1-2 System Model and Assumptions . . . 36
4-2 Training Results . . . 37
4-2-1 Hyperparameter Configuration and Training Process . . . 37
4-3 Prediction Accuracy of Transformer-Based MPC in Open Loop . . . 41
4-3-1 Experimental Setup . . . 42
4-3-2 Performance Evaluation of Transformer-Based MPC . . . 43
4-4 Closed Loop MPC Simulations . . . 49
4-5 Comparison of Transformer-Based MPC Against Benchmark LSTM . . . 51
4-6 Discussion . . . 53
4-7 Conclusion . . . 55

5 Conclusion and Future Work 57
5-1 Contributions . . . 57
5-2 Conclusion . . . 58
5-3 Recommendations for Future Work . . . 59

A Binary Decision Conversion Data Preprocessing 61

B Computational Optimisation: ONNX Runttime 63

C System Equations 65
C-1 Energy Storage System (ESS) . . . 65
C-2 Variables . . . 65
C-3 MLD Constraints . . . 66

D Paper 69

Bibliography 77

Glossary 81
List of Acronyms . . . 81
List of Symbols . . . 81

Acknowledgements and Preface

Dear reader,

Now that my thesis has come to an end, so too does my time as a student. This thesis was the largest and longest project I have undertaken, making it a truly interesting journey. As someone relatively new to the machine learning world, there was a lot to learn about the subject. I am pleased to report that my interest in the topic has only grown over the past year. During this time, I have gained a considerable amount of knowledge not only about machine learning in control but also about the workings of academic research. This year has been a period of significant personal and professional growth.

First of all, I want to thank Caio Fabio and Bart. Caio Fabio, I thoroughly enjoyed our weekly meetings; I admire your patience. You always supported my ideas, offered sound advice, and steered me in the right direction numerous times. I wish you all the best for your PhD in the coming years. Bart, I am amazed by your knowledge and have learned so much during our monthly meetings. You could always pinpoint crucial elements during the meetings, providing excellent advice and guidance.

Furthermore, I want to thank "de systeembeheerders", with whom I studied (and had coffee breaks) a lot during my thesis and the rest of my master's program. I want to extend particular thanks to Emiel and Berend, without whom I do not think I could have completed all the projects during my master's.

Last but not least, a massive thank you to my family and friends for always being there to listen to yet another story about transformers, which I am sure was more beneficial for me than for them. Your patience and willingness to lend an ear have been a huge support. I could not have done this without your encouragement and love.

Thank you all,

Bob

Chapter 1

Introduction

1-1 Background

As global energy demands continue to rise and environmental sustainability becomes an increasingly critical concern, microgrids have emerged as a key solution within the energy sector [1],[2]. These decentralised networks, capable of operating both independently and in conjunction with the traditional centralised electrical grid, excel in integrating renewable energy sources such as solar and wind power, thereby enhancing grid resilience and minimising carbon footprints. The significance of these systems is underscored by a report from the International Renewable Energy Agency (IRENA), which suggests that renewable energy could potentially meet up to two-thirds of the total global energy demand [3]. The capacity of microgrids for high renewable energy integration is crucial for reducing greenhouse gas emissions. It is instrumental in global efforts to meet temperature increase targets set by the Paris Agreement on climate change [4]. Additionally, microgrids ensure energy security and reliability, especially in areas susceptible to extreme weather or those geographically isolated from central power grids [5]. Overall, by minimising dependencies on centralised power supplies, microgrids support the stability of the power grid and contribute to reducing global carbon emissions.

A significant challenge within microgrid management is the unit commitment problem. This complex mathematical optimisation task involves coordinating a group of energy production units to meet demand at minimal cost [6]. In this context, model predictive control (MPC) can be a practical approach to managing the complexities of microgrids. MPC is particularly valued for its predictive nature [7]. It bases the control strategy on future system behaviours and predictions, which are crucial in environments that greatly depend on demand forecasts and variable renewable energy generation. Furthermore, MPC relies on a feedback mechanism that enhances system resilience against uncertainties, making it highly effective in environments with unpredictable energy supply. Also, MPC is capable of handling various power systems constraints such as generator capacity and ramp rate limits, ensuring the microgrid operates within its optimal parameters while maintaining safety and efficiency [8],[9]. These features of MPC make it a highly suitable choice for addressing the unit commitment problem within microgrids. It involves the complex coordination of energy production units to meet demand efficiently and cost-effectively. The inherent interaction between continuous and discrete variables in microgrids is well-modelled by mixed logical dynamical (MLD) systems and formulated as mixed-integer linear programming (MILP) problems within the MPC framework [9]. By optimising operations based on current conditions and anticipated future scenarios, MPC can significantly enhance the microgrids' economic and environmental performance.

However, the application of MPC in microgrids faces significant challenges due to the computational intensity of solving MILP problems at each control iteration, which is time-consuming and computationally demanding. To address this challenge, researchers have adopted learning-based methods to enhance the online applicability of MPC in microgrids, e.g. [10]. Our research explicitly explores the use of transformers—a machine learning model renowned for recognising patterns in sequential data and capability for parallel processing [11]. The decision-making process in MPC is inherently a sequence modelling problem, as control actions are based on past and present data, which aligns well with the capabilities of transformers to process such sequences efficiently. By learning and predicting the binary decisions in MILP problems, transformers can potentially simplify these decisions to more tractable linear programming problems, thereby enhancing the real-time applicability of MPC in microgrid control.

1-2 Problem Description

The unit commitment problem within microgrids is a complex mathematical optimisation challenge that requires the coordination of various energy production units to meet energy demands efficiently while minimising operational costs. This problem is particularly challenging due to the hybrid nature of the energy resources involved, which includes both continuous variables, such as power outputs, and discrete decisions, such as generators' on/off status.

This research identifies MPC as an effective control strategy for addressing the unit commitment problem. MPC's ability to predict future system states and its robust handling of multivariable constraints make it highly suited for managing the intricacies of hybrid systems. However, the effectiveness of MPC relies on the solution of MILP problems at each iteration of the control process. These MILP problems, essential for determining the discrete decisions inherent in microgrid operations, are computationally intensive and belong to the class of $\mathcal{NP}$-hard problems, posing significant challenges for real-time applications [12].

To overcome these challenges, this research proposes a novel approach by integrating transformers into the MPC framework. Transformers, known for their exceptional capability to process sequential data through self-attention mechanisms, offer a promising solution for learning and predicting the binary decisions required in MILP. This strategy aims to transform the computationally demanding MILP problems into more tractable linear programming (LP) problems, thereby reducing the computational burden and enhancing the scalability and speed of MPC implementations in real-time microgrid control.

This innovative approach leverages the parallel processing capabilities of transformers, which could lead to more efficient training, making it feasible to handle more extensive and complex microgrid systems data. The proposed research explores whether transformers can maintain the decision-making accuracy of MPC while significantly reducing the computational load involved in solving MILP problems. The following research question is addressed in this research:

How can transformers be applied to predict the optimal binary solution of Mixed-Integer Linear Programming problems, specifically within the context of Mixed Logical Dynamical systems and learning-based Model Predictive Control?

This study will first assess the suitability of the transformer architecture for this unique application to address the question. It is expected that the transformer model needs to be adapted from its original use in natural language processing to suit the specific characteristics of microgrid data. The effectiveness of the transformer-based MPC model is evaluated against existing techniques by comparing training complexity and operational performance in simulations. This report details the methodologies employed and outlines the structure of the investigation, setting the stage for an in-depth exploration and analysis.

1-3 Thesis Outline

The structure of the thesis is given in this section. Chapter 2 provides an in-depth background and literature review that explores the foundational concepts of microgrids and the challenges of controlling these systems. It elaborates on the system model used for microgrids, discusses the pivotal role of MPC, and dives into the complexities of solving MILP problems. The literature on learning-based methods within MPC is reviewed, highlighting the innovative potential of transformers in this field.

Following the foundational background, Chapter 3 progresses into the methodology, where the approach to implementing transformer models for binary sequence prediction in MILP problems is detailed. This section covers everything from data collection and preprocessing to the specific integration steps with MPC frameworks and the evaluation metrics necessary for validating the methodology.

In Chapter 4, the practical application of the transformer-based MPC in microgrid control is examined through detailed simulations. This part describes the experimental setup and the assumptions made. It also critically analyses the performance results of the transformer model, discussing the implications and challenges encountered during the simulations.

The thesis concludes in the Conclusion and Future Work Chapter 5, which summarises the insights gained through the research and reflects on the contributions made. It outlines the potential directions for future research that could further refine the use of advanced machine-learning techniques in the realm of MPC for microgrids.

Supplementary details, including mathematical formulations, elaborations and a paper draft on the topic, are provided in the Appendices. The codes used in this thesis are provided in [13].

Chapter 2

Background and Literature Review

This chapter provides an overview of the complexities inherent in microgrid systems and the advanced control strategies essential for their effective management. It introduces mixed logical dynamical (MLD) systems, which model the interplay between continuous and discrete variables within microgrids. The control problem within these MLD systems can be effectively formulated as mixed-integer linear programming (MILP) problems, enabling the use of model predictive control (MPC) to determine optimal control strategies.

However, implementing MPC in real-time is challenging due to the combinatorial nature of MILP problems, which involve complex binary decision-making processes. These computational challenges make online MPC challenging, impacting its effectiveness and efficiency in operational scenarios.

This chapter delves into integrating machine learning techniques within the MPC framework to overcome these computational challenges to enhance real-time optimisation capabilities. Exploring existing machine learning applications in similar contexts provides insights into the potential of learning-based MPC, encouraging further investigation into other advanced learning-based techniques. Specifically, it underscores the potential of transformers, a sequence modelling technique renowned for its capabilities in handling sequential data, to improve predictive accuracy and computational efficiency. By learning the binary components of MILP formulations in hybrid MPC, transformers could significantly enhance the scalability and practicality of deploying MPC in real-time scenarios.

2-1 System Model: Microgrids

This section describes the modelling of microgrids, which is central to this thesis. It discusses its main components and explains how to model and control them effectively. Microgrids represent an evolution in electricity distribution networks, aiming to sustainably address the increasing energy demand. They are essentially subsystems of the distribution grid, incorporating generation capacities, storage devices, and controllable loads. These elements work in unison to operate as a single controllable entity that can be connected to or isolated from the utility grid. The operation of microgrids introduces a complex interplay of various components and dynamics, necessitating advanced control strategies to optimise their performance effectively.

The significance of microgrids lies in their capability to enhance the decentralised management of distributed energy resources, including distributed generators and renewable energy sources, thereby reducing the reliance on the centralised grid. This decentralised approach promotes the efficient use of renewable energy and improves the resilience of the electricity supply, e.g. in grid outages or peak demand periods [9].

One of the main challenges in microgrid management is the unit commitment problem. This complex mathematical optimisation task involves coordinating a group of energy production units to meet demand at minimal cost [6]. This is particularly challenging due to the inherent uncertainty in energy demand, renewable energy generation, and energy prices. These uncertainties require advanced modelling and control strategies to anticipate future conditions and adjust operations accordingly to maintain stability and efficiency. MPC emerges as a highly effective solution in this context, with its ability to anticipate predictions of future system behaviour to optimise microgrid operations [9]. Section 2-1-2 delves deeper into the MPC problem of microgrids. First, a general system description of the microgrid is given. This system description is based on the elaborate system description of microgrids in [10] and [9]. This description is used as the basis throughout this thesis report.

2-1-1 System Description

This section outlines the problem of power dispatching in a microgrid environment, integrating storage elements (e.g., batteries and ultracapacitors), local generators, a bidirectional connection to the main grid, and uncontrollable loads.

Dynamic Modeling of Energy Storage Systems

In this study, we limit our analysis to microgrids where the energy storage system is modelled as a battery that adheres to the following hybrid dynamical law, earlier described in [10]:

$$x_{\mathrm{b}}(k+1) = \begin{cases} x_{\mathrm{b}}(k) + \frac{T_{\mathrm{s}}}{\eta_d} P_{\mathrm{b}}(k) & \text{if } P_{\mathrm{b}}(k) < 0 \\ x_{\mathrm{b}}(k) + T_{\mathrm{s}} \eta_{\mathrm{c}} P_{\mathrm{b}}(k) & \text{if } P_{\mathrm{b}}(k) \geq 0 \end{cases} \tag{2-1}$$

where $x_{\mathrm{b}}(k)$ represents the energy level at step k, η_{c} and η_d denote charging and discharging efficiencies, $P_{\mathrm{b}}(k)$ is the power exchanged with the energy storage system, and T_{s} is the sampling interval.

Using an MLD approach, we introduce a binary variable $\delta_{\mathrm{b}}(k)$ to indicate the storage system's mode (charging or discharging), enabling a compact representation exploitable for control applications [10]:

$$x_{\mathrm{b}}(k+1) = x_{\mathrm{b}}(k) + T_{\mathrm{s}} \left(\eta_{\mathrm{c}} - \frac{1}{\eta_{\mathrm{d}}} \right) z_{\mathrm{b}}(k) + \frac{T_{\mathrm{s}}}{\eta_{\mathrm{d}}} P_{\mathrm{b}}(k), \tag{2-2}$$

with $\delta_{\text{b}} = 1 \iff P_{\text{b}}(k) \geq 0$ denotes charging mode or vice-versa discharging mode. Further, the term $z_{\text{b}}(k) = \delta_{\text{b}}(k)P_{\text{b}}(k)$ modulates the effect of power exchange on the energy level, depending on the mode of operation. The logical correlation between $\delta_{\text{b}}(k)$ and $P_{\text{b}}(k)$, along with the formulation of $z_{\text{b}}(k)$, can be reformulated into a series of mixed-integer linear inequalities as described in [14].

Generator Units

The microgrid features two types of generation units: renewable sources (zero-cost, uncontrollable power) and dispatchable generators, which can be controlled within bounds and are therefore considered control variables. The output power of dispatchable units, represented by $\boldsymbol{P}_{\text{dis}}(k)$, is adjustable. Let $\boldsymbol{P}_{\text{dis}}(k)$ be the vector representing the power produced by dispatchable generators, defined as

$$\boldsymbol{P}_{\text{dis}}(k) = \begin{bmatrix} P_1^{\text{dis}}(k) \\ \vdots \\ P_{N_{\text{gen}}}^{\text{dis}}(k) \end{bmatrix}, \tag{2-3}$$

where $P_i^{\text{dis}}(k)$ denotes the power produced by dispatchable unit i at time step k, and N_{gen} is the total number of dispatchable units. Binary variables are used to determine which dispatchable generators are turned on or off ($\delta_i^{\text{on}} = 1$, and $\delta_i^{\text{off}} = 0$).

Prices of Energy

The energy flows within the microgrid have different associated energy costs. We assume three types of prices: $c_{\text{buy}}(k)$ for purchases, and $c_{\text{sale}}(k)$ for sales from and to the main grid. Further, $c_{\text{prod}}(k)$ is the cost for local energy generation with the dispatchable units. It is further assumed that estimates of these prices are known over a certain horizon in the future.

Grid Interaction

The microgrid can supplement the central grid or operate independently, engaging in energy transactions dictated by demand, generation capacity, and economic considerations. These transactions are captured through the power exchange variable $P_{\text{grid}}(k)$, with the binary decision variable $\delta_{\text{grid}}(k)$ indicating the microgrid's operational state relative to the main grid—either importing ($\delta_{\text{grid}}(k) = 1$) or exporting energy ($\delta_{\text{grid}}(k) = 0$). The economic impact of these transactions, whether cost or revenue, is modelled by $C_{\text{grid}}(k)$:

$$C_{\text{grid}}(k) = \begin{cases} c_{\text{buy}}(k) \cdot P_{\text{grid}}(k) & \text{if } \delta_{\text{grid}}(k) = 1 \text{ (importing)} \\ c_{\text{sell}}(k) \cdot P_{\text{grid}}(k) & \text{if } \delta_{\text{grid}}(k) = 0 \text{ (exporting)} \end{cases} \tag{2-4}$$

where $\delta_{\text{grid}}(k) = 1$ represents the importing case (buying energy) and $\delta_{\text{grid}}(k) = 0$ denotes the exporting case (selling energy). The decision to import or export is determined by current demand, generation capacity, and economic factors. This model ensures the microgrid's operations align with both internal needs and external market conditions, optimising the economic benefits of grid interaction.

2-1-2 Control of Microgrids using MPC

MPC utilises a model-based strategy, using predictions about system behaviour and inputs to determine optimal control actions over a defined horizon. Many research efforts have focused on applying MPC to scenarios involving both continuous and discrete constraints [14]. This exploration has naturally extended MPC's application to hybrid systems characterised by the interaction between continuous dynamics and discrete events. Such advancements have broadened MPC's applicability across various domains, including the complex operational dynamics of microgrids.

The adaptation of hybrid MPC formulations for microgrids is well-established in the literature, with notable successes in, for example, [9], [15] and [16]. These successes underscore the methodology's ability to balance operational constraints with cost optimisation, giving it a compelling strategy for microgrid control. Given the inherent continuous and discrete variables within microgrid systems, framing the microgrid control problem within a hybrid MPC formulation is a reasonable choice.

As observed in 2-1-1, the dynamics within the microgrid are well-suited for representation in an MLD formulation. This formulation can seamlessly be translated into a MILP problem and function as constraints in the hybrid MPC formulation. This conversion facilitates the practical application of MPC to the microgrids, ensuring both operational integrity and cost efficiency.

To formulate the MPC problem, we first define the objective function. The objective is to minimise economic costs while satisfying load demands and operational constraints. The cost function comprises economic costs from local energy production via dispatchable units and the energy exchanged with the main grid. The objective function is formulated as:

$$\min_{\boldsymbol{P}_{\text{dis}}(k), P_{\text{grid}}(k), P_{\text{b}}(k), \boldsymbol{\delta}(k), \boldsymbol{z}(k)} J(\boldsymbol{P}_{\text{dis}}(k), C_{\text{grid}}(k), c_{\text{prod}}(k)) \tag{2-5}$$

where the cost function is defined as:

$$J(\boldsymbol{P}_{\text{dis}}(k)), C_{\text{grid}}(k), c_{\text{prod}}(k) = \sum_{j=0}^{N_{\text{p}}-1} (C_{\text{grid}}(k+j) + c_{\text{prod}}(k+j) \sum_{i=1}^{N_{\text{gen}}} P_i^{\text{dis}}(k+j)) \tag{2-6}$$

where N_{p} is the prediction horizon. The objective function is subjected to the constraints of the microgrid. The system dynamics of the microgrid model can be represented in the MLD formulation as mixed integer constraints. These are formulated in [10] as follows:

$$
\begin{aligned}
& E_1\boldsymbol{\delta}(k) + E_2\boldsymbol{z}(k) \leq E_3\boldsymbol{u}(k) + E_4 && \text{(2-7a)}\\
& P_{\mathrm{b}}(k) = \sum_{i=1}^{N_{\mathrm{gen}}} P_i^{\mathrm{dis}}(k) + P_{res}(k) + P_{\mathrm{grid}}(k) - P_{load(k)} && \text{(2-7b)}\\
& \underline{P}_{\mathrm{b}} \leq P_{\mathrm{b}}(k) \leq \bar{P}_{\mathrm{b}} && \text{(2-7c)}\\
& \underline{P}_{\mathrm{grid}} \leq P_{\mathrm{grid}}(k) \leq \bar{P}_{\mathrm{grid}} && \text{(2-7d)}\\
& \delta_i^{on}\underline{P}_{\mathrm{grid}} \leq P_{\mathrm{grid}}(k) \leq \delta_i^{on}\bar{P}_{\mathrm{grid}} && \text{(2-7e)}\\
& \underline{x}_{\mathrm{b}} \leq x_{\mathrm{b}}(k) \leq \bar{x}_{\mathrm{b}} && \text{(2-7f)}
\end{aligned}
$$

$$
\begin{aligned}
& \boldsymbol{u}(k) = [\boldsymbol{P}_{\mathrm{dis}}^{\top}(k), C_{\mathrm{grid}}(k), P_{\mathrm{b}}(k)]^{\top}\\
& \text{ for } i = 1, \ldots, N_{\mathrm{gen}}\\
& \text{ for } k = 0, \ldots, N_{\mathrm{p}} - 1
\end{aligned}
$$

constraint (2-7a) follows from the earlier described MLD formulations of the microgrid. Constraint (2-7b) ensures power balance by specifying that all generated power must be consumed within the system, stored, or sold to the main grid at each time step. Constraints (2-7c) to (2-7f) define physical constraints on components of the microgrids: (2-7c) constrains the power exchanged with the battery, (2-7d) the power exchange with the main grid, (2-7e) the produced power production per power unit and (2-7f) defines the physical bounds on the state of charge of the battery.

The MPC-MLD formulation is a promising approach to control the microgrid system effectively. However, applying the MPC in MLD systems remains challenging. As each iteration requires solving an MILP problem, as described above. MILP problems are combinatorial problems in nature, with computational complexity growing exponentially as the number of discrete decision variables increases. Hence, MILP problems belong to the $\mathcal{NP}$-hard class, which means that there are no algorithms known to guarantee the solution of the problem within polynomial time [12]. Therefore, several recent works developed approaches to overcome this issue of computational complexity to make MPC applicable to MLD systems. Particularly, learning-based approaches have demonstrated great success over the past few years in hybrid MPC, e.g. [17], [18], [19]. The upcoming Section 2-2 gives an overview of research works that addressed this challenge.

2-2 Related Work

This section overviews recent approaches to solving the challenge of solving MILPs for MPC in an online setting. Section Section 2-2-1 first briefly discusses the problem and challenges of MILP, after which Section 2-2-2 discusses research works that integrated machine learning and MPC to overcome the challenges inherent in hybrid MPC.

2-2-1 Mixed-Integer Linear Programming

Mixed-integer linear programming is a mathematical optimisation technique that solves problems involving continuous and discrete decision variables. The main goal of an MILP problem is to optimise a given linear objective function while satisfying a set of linear constraints [20]. Effectively solving MILP problems is crucial in numerous applications, e.g., finite horizon optimal control of hybrid systems. However, since MILP problems involve discrete decision variables, the problem becomes combinatorial. In combinatorial problems, the size of the problem grows exponentially with an increasing number of discrete (decision) variables, making them belong to the $\mathcal{NP}$-hard class [20].

Due to several factors, MILP problems are more complex than linear programming (LP). Firstly, the presence of discrete decision variables transforms MILP into a combinatorial problem, where the objective is to find the optimal arrangement or selection of objects from a finite set, making the problem inherently complex as the problem size increases [21]. Secondly, MILP's integer variables make the problem non-convex, complicating the search for a global optimum compared to the convex nature of LP problems, where efficient algorithms exist for finding the global optimum. Furthermore, the combinatorial and non-convex characteristics of MILP require specialised algorithms, such as branch-and-bound, cutting plane methods, or heuristics, to navigate these challenges. These challenges are further compounded by the exponential growth in complexity with increased problem size, emphasising the need for efficient, sometimes problem-specific, algorithms to manage computational demands while maintaining solution quality [20]. Machine learning introduces a new dimension to this landscape by offering simplifications or speed improvements through machine learning. An example of this is demonstrated in [17], where machine learning is applied to warm-start branch and bound, which achieves significant speed improvements without compromising solution quality.

In optimal control, MILP problems must be solved within time limits to make the controller applicable in online settings. Recent research has explored leveraging machine learning to overcome these challenges, given the computational challenges associated with MILP in real-time MPC applications. The following section delves into how supervised and reinforcement learning techniques can be applied to streamline MPC operations, thereby enhancing the efficacy and responsiveness of microgrid control.

2-2-2 Learning-based MPC

Incorporating learning methodologies within MPC marks a considerable progression in control systems, especially for those defined by MLD systems. This integration facilitates the application of MPC in online scenarios, which is a challenge due to the complexity and computational demands of solving MILP problems inherent in MLD systems. The literature reviewed outlines two effective learning-based approaches: supervised learning-based MPC and reinforcement learning-based MPC, referred to as SL-based MPC and RL-based MPC.

Reinforcement Learning Based MPC

Reinforcement learning in control is the subfield of machine learning that studies how past data can be used to enhance future manipulation of a dynamical system by an agent [22].

The agent interacts with its environment and receives feedback through rewards and punishments, which it uses to learn a policy that maps states to actions. Two key aspects of RL are exploration and exploitation. Exploration involves the agent trying out new actions to discover their outcomes. In contrast, exploitation consists of choosing actions that the agent believes will yield the highest rewards based on current knowledge. This approach allows for continual improvement in handling complex, variable environments, as demonstrated in various applications documented in the literature [23].

In RL-based MPC has been applied in several ways. RL has been used to learn and improve the model used in MPC. Learning a better and more accurate model effectively enhances the performance of the MPC-controlled system. A more precise model will lead to more accurate predictions of the system's future state, allowing the MPC to make better decisions about future control actions. Another application of RL in combination with MPC can also be to speed up calculations over the horizon of MPC. In this case, RL can approximate the optimal control policy, which can then be used to quickly compute the control actions for each time step. This can be a significant advantage for MPC problems with a long horizon, as the computational time required to solve the MPC problem can be significantly reduced or when online MPC requires fast solutions [22]. In many applications, it is advantageous to use MPC as a primal controller and RL as a secondary controller since one can exploit guarantees on safety and stability with MPC in these scenarios [24]. However, using RL to take over the MPC as primal controller is also possible. Here, the use of RL in combination with MPC can lead to stability and safety challenges since the main objective in the field of RL, i.e. optimising the reward function, is not primarily similar to the main objectives in control, i.e. stability and safety. Integrating MPC into RL practices allows for enforcing stability and adherence to constraints. This integration helps direct the learning process towards feasible and safe policies. This approach leverages the structured control strategy of MPC, with its established ability to handle dynamic constraints and predict system behaviour accurately, to inform and guide the learning algorithms of RL [24].MPC can benefit from the contrasting properties of RL; RL is model-free, adaptive, and has low online complexity. On the other hand, RL can also benefit from the properties and majority of knowledge connected to MPC in terms of stability, feasibility and robustness theory. RL encounters stability and safety challenges, particularly when agents need to interact with real-world systems [24].

Integrating RL methodologies into MPC has led to innovative developments. The work presented in [25] proposes an RL-enhanced MPC approach to leverage RL's adaptability while addressing its safety challenges using robust MPC as a safety mechanism. This approach effectively balances performance improvement with safety guarantees despite potential limitations in exploration, conservatism, and controller generalisability.

Similarly, [26] introduces a Predictive Safety Filter to ensure safety in RL-based control inputs, allowing RL more freedom in operation by filtering out unsafe suggestions. This method combines MPC's reliability with RL's adaptability, though it introduces additional complexity and demands careful coordination.

Contrastingly, [27] focuses on simplifying MPC through RL by optimising the prediction horizon, demonstrating significant performance improvements in simpler systems such as inverted pendulums. This approach highlights the potential for efficiency and challenges in applying it to more complex systems and ensuring safety and stability.

These studies illustrate the potential of RL-based MPC approaches to enhance optimal control systems, giving a blend of performance, adaptability, and safety. However, the complexities and computational challenges of implementing these strategies in practical, real-world applications should be acknowledged. Further, in this thesis, the main emphasis is on simplifying the MILP problems to make MPC more applicable in online applications. Applications in supverised learning (SL)-based MPC have shown more successful techniques for accomplishing this.

Supervised Learning based MPC

The literature on SL-based MPC contains several examples that simplify the hybrid MPC problem, improving its applicability in online environments. Supervised learning is a machine learning framework where models are trained to recognise patterns and relationships within data based on labelled examples. The dataset for training includes input data paired with corresponding output labels, serving as the ground truth. This training process aims to establish a mapping from inputs to outputs, enabling the model to predict outcomes accurately on new, unseen data. In the context of hybrid MPC, particularly discussed in 2-2-1, the predictive power of SL could simplify the challenging part of MILP problems. Researchers have explored whether SL can learn and predict the binary variables in MILP offline, transforming the problem's complexity into a computationally tractable LP problem. This adaptation holds significant potential for enhancing the online applicability of MPC [28]. It offers the possibility, though not a guarantee, that the controller will function effectively online. Several research examples underscore the potential of such approaches.

The SL-based controllers presented in [12], [29] and [18] exemplify a common strategy for achieving efficient online control of MPC problems in MLD systems, where MILP problems are inherently involved. By solving several MILP instances offline, these approaches leverage SL and classification techniques to identify and simplify the binary components of the problems. This simplification transforms complex MILP problems into more manageable LP problems or reduces the number of binary variables involved, significantly reducing the computational time required for online applications.

While promising, the researchers highlight a crucial trade-off between solution optimality and computational complexity. While the SL-based controllers do not provide strict optimality guarantees, they significantly reduce the computational burden, enhancing such methods' practical utility. This trade-off is vital in real-world scenarios where the need for timely and adaptable decision-making often outweighs the pursuit of theoretical perfection.

Another essential consideration for the effectiveness of these SL-based MPC controllers is the balance between the volume of training data and the computational resources available. Although a more extensive dataset can lead to more reliable predictions, it also demands greater computational power. Data collection and binary variable prediction typically occur offline, mitigating potential issues during online operations. However, as computational capabilities continue to evolve, the impact of data volume on computational resources is expected to lessen.

Interestingly, current SL-based MPC methods often overlook the inherent temporal structure of MPC sequences. Given that the primary objective of SL-based MPC is to forecast binary sequences resulting from system decision-making—influenced by previous and future decisions—it stands to reason that recognising and leveraging this temporal structure could be beneficial. The sequential nature of MPC typically generates patterns in binary inputs, offering critical insights that could significantly enhance solution techniques. Techniques like long short-term memory (LSTM)s have demonstrated potential in exploiting these temporal dynamics, as shown in [18]. However, recent advancements in sequence modelling have introduced new architectures, notably transformers, which have surpassed the performance of LSTMs in various domains. Given their proven efficacy in handling sequential data, exploring whether transformers could similarly enhance SL-based MPC would be interesting. This will be further investigated in the upcoming chapters.

2-2-3 Conclusions from the Literature

This literature review highlights the significant challenges and advancements in applying MPC in MLD systems, specifically microgrids. It has delved into key strategies such as supervised learning and reinforcement learning, which enhance MPC's real-time applicability. While RL offers adaptability and potential speed enhancements, it raises concerns about stability and safety. Conversely, SL-based methods have proven effective in simplifying the complex components of MILP, rendering MPC more computationally tractable in online environments. These methods notably reduce the computational burden by learning to predict the binary decision variables, thus enhancing the system's operational performance.

The review also highlights a research gap: current SL-based MPC methods often do not take into account the inherent temporal structure of MPC sequences. Integrating advanced sequence modelling techniques can possibly further enhance SL-based MPC. While LSTMs have shown utility in capturing temporal dynamics within MPC, transformers have demonstrated superior performance in various other fields by effectively handling sequential data. This suggests their potential for enhancing MPC, which will be a focus of further investigation in this thesis.

2-3 Transformers

The evolution of sequence modelling techniques has been crucial in advancing machine learning applications, particularly in natural language processing (NLP) and handling time-series data. While Recurrent Neural Networks (RNNs) and long short-term memory LSTM networks marked significant milestones by effectively capturing temporal dependencies, they faced limitations in parallel computation and long-range dependency modelling [30]. These challenges prompted the development of the transformer architecture, which has significantly influenced the landscape of deep learning.

The transformer architecture was first introduced in the seminal paper "Attention is all you need" [11]. Unlike traditional sequence models that rely on recurrent operations, the transformer eliminates the need for sequence-aligned operations. Transformers rely on a mechanism known as self-attention to compute representations of its input and output. This shift facilitates parallel processing and enhances efficiency in handling sequences. The architecture consists of two main components: an encoder that processes the input sequence and a decoder that generates the output sequence. For a visual representation of this architecture, we refer to Figure 2-1, which schematically outlines the encoder-decoder structure and its components. The figure is based on the figure presented in [11].

Self-Attention Mechanism

The core component of the transformer model is the self-attention mechanism, which allows the model to dynamically evaluate and assign relevance to different parts of the input sequence for output generation. This mechanism significantly improves the learning of long-range dependencies compared to RNNs and LSTMs, where the computation of each output element depends sequentially on previous elements, a process prone to vanishing and exploding gradients.

The self-attention process involves computing three vectors for each token in the input: a query vector, a key vector, and a value vector. These vectors are derived through training and are essential for attention computation. The process computes the attention as a weighted sum of the values, where the weights are determined by the compatibility of the query with the corresponding keys:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{2-8}$$

here $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ represent the matrices formed by stacking the query, key, and value vectors of all input tokens, respectively. Further, $\sqrt{d_k}$ is a scaling factor that helps stabilise the learning process. This scaled dot-product attention is central to the transformer's ability to prioritise information from different parts of the input sequence, regardless of their positions.

The self-attention mechanism improves dependency recognition and facilitates parallel processing of data, a significant difference compared to previous sequence modelling techniques. This parallelism is critical for the transformer's ability to process all sequence parts simultaneously, dramatically boosting computational efficiency and making the architecture well-suited for handling large datasets.

Furthermore, multi-head attention enables running several attention processes in parallel, each with its own set of parameters, further enhancing the architecture. Multi-head attention, also shown in Figure 2-1, allows the model to simultaneously capture various relationships from different representational spaces at various sequence positions. This capability boosts the model's accuracy and flexibility and enhances its ability to comprehensively address multiple aspects of the input data, thus improving the overall effectiveness of the attention process.

Multi-Head Attention

The multi-head attention mechanism, a central component of the transformer architecture, is vital for handling the complex, multi-faceted data streams in microgrid control. This mechanism has been designed to enhance its ability to process and interpret the temporal and spatial relationships within our application of microgrid data, ensuring the system can effectively respond to real-time demands and forecast future needs.

In the dynamic environment of microgrid control, data elements such as energy prices, power demands, and system states are interdependent and exhibit complex patterns of interaction. The multi-head attention mechanism navigates these complexities by allowing the model to focus on different data segments concurrently. This capability is critical for understanding how various variables influence each other over time and under different conditions.

The multi-head attention allows the transformer to handle the various aspects of the microgrid source data, such as energy consumption patterns, pricing statistics and the current battery state of the system, by focusing on different segments of the data concurrently. This enables the model to understand the deeper relationships between all aspects of the data and how the decisions within the MPC arise.

Positional Encoding

Another crucial component of the transformer model is positional encoding, which compensates for the architecture's lack of inherent sequence processing capability. Unlike RNNs and LSTMs, transformers do not process data sequentially by default. Therefore, they require a method to incorporate information about the order of the sequence. Positional encodings are added to each input embedding to provide this sequence context. The positional encoding function often uses sine and cosine functions of different frequencies [11]:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/1000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/1000^{2i/d_{model}}) \end{aligned} \tag{2-9}$$

in this formula pos is the position within the sequence and i is the dimension index. Each dimension of the positional encoding corresponds to a sinusoidal function, allowing the model to learn relative positions.

Additional Architectural Components

Each transformer layer features a feed-forward neural network that applies the same linear transformation to each position. This uniform processing ensures consistent transformation across all sequence positions. The architecture also incorporates residual connections and layer normalisation (referred to as 'Add & Norm') after each sub-layer, which are critical for maintaining training stability and preventing performance degradation in deeper network structures.

The initial step in processing involves embedding the input tokens and converting discrete input features into dense vectors that are optimally formatted for subsequent processing by the model's neural layers. These embeddings are especially crucial in transforming the input into a structure amenable to the self-attention layers. This step is particularly relevant in language models, which were the primary focus of the original transformer paper [11].

Applications and Implications

Transformers have rapidly become a dominant architecture in sequence modelling, particularly within natural language processing (NLP). They have set benchmarks in machine translation, text summarisation, and sentiment analysis tasks. Moreover, transformers have extended their applicability beyond NLP to fields like image recognition. For instance, the Vision Transformer approaches image analysis by treating images as sequences of patches, applying Transformer encoders in a method analogous to text processing [31]. This strategy has shown

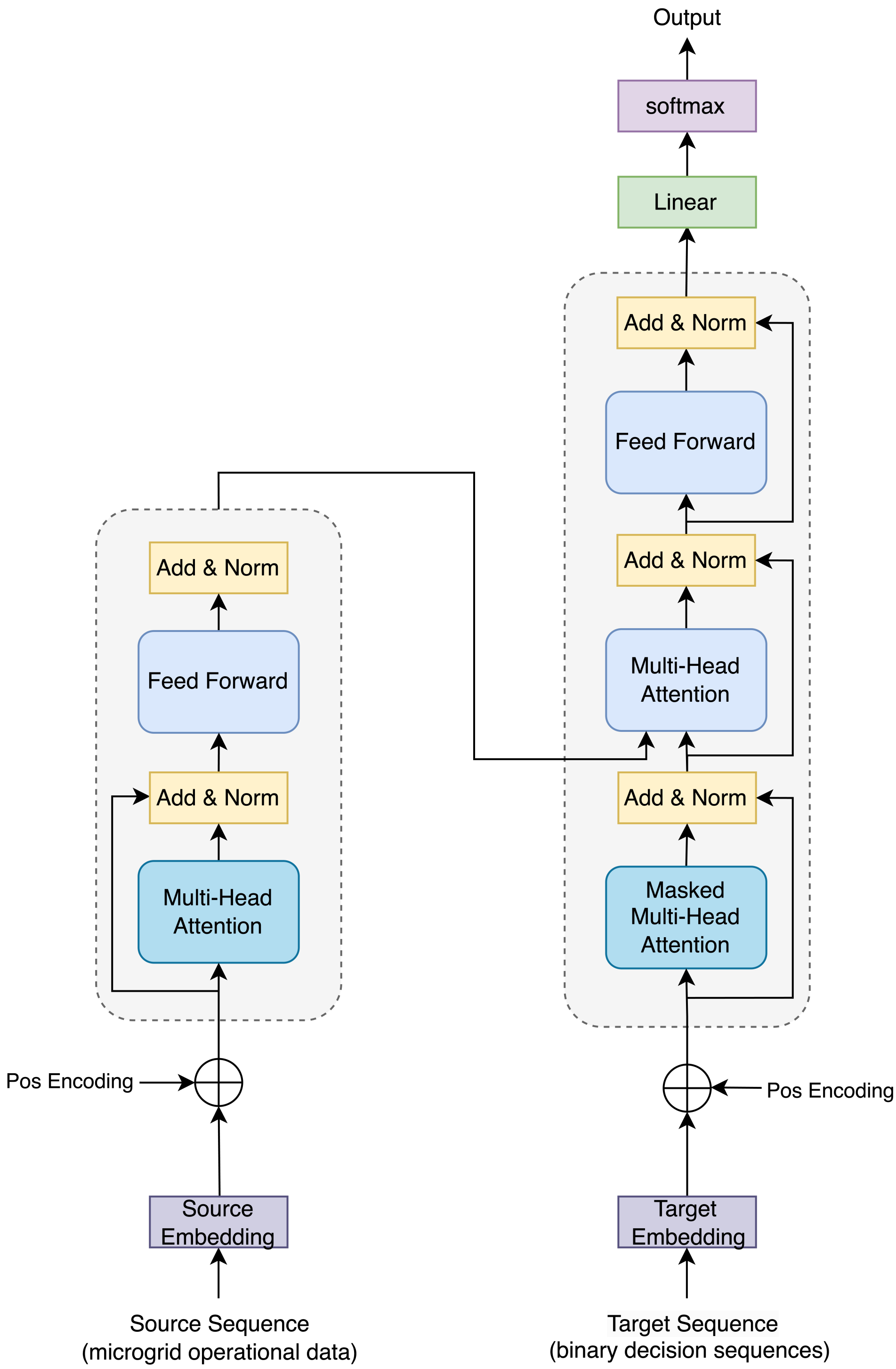

Figure 2-1: Transformer architecture Diagram, based on [11]

remarkable success, achieving or surpassing the best performances in image classification tasks. This adaptability and robust performance underline the potential of transformers to revolutionise other fields, including potentially enhancing MPC systems by simplifying MILP problems, which is the main focus of this thesis.

Motivations for Using Transformers in Microgrid MPC

The implementation of transformers in MPC for microgrids has several motivations. First, transformers, known for their parallel processing capabilities, offer a promising solution to the computational challenge in microgrids. By enabling fast and accurate computations, transformers can significantly reduce the time required for training and decision-making processes, which is essential for the dynamic operating environments of microgrids.

Secondly, research has shown the great potential of transformers in different fields regarding prediction accuracy compared to other learning-based techniques. The self-attention mechanism in transformers allows for a more nuanced understanding of long-range dependencies within data sequences. This could lead to better forecasting accuracy of load demands and resource availability, thus optimising operational strategies.

Further, since the primary goal of SL-based MPC is to forecast binary sequences that emerge from decision-making, which are influenced by past and anticipated future decisions. Recognising and harnessing this temporal structure is critical. The sequential nature of MPC can be assumed to reveal patterns in binary inputs that provide essential insights. Transformers, known for their excellence in sequence modelling, are considered to be well-suited for this task.

Lastly, the transformer architecture is well suited to scale for larger problems. This scalability makes transformers ideal for application in growing microgrid networks.

2-4 Summary

This chapter overviews the operational complexities and control challenges inherent in microgrid systems. By using MLD systems, we can effectively model the interplay between continuous and discrete variables that characterise the dynamic environment of microgrids. The discussion underscored the pivotal role of MPC in controlling these systems through advanced optimisation techniques, specifically MILP.

While MPC provides optimal control solutions for MLD systems, its online application often faces significant challenges due to the computational demands of solving MILP problems. These challenges arise primarily from the combinatorial nature of MILP in the MPC problem. This becomes particularly relevant in microgrids, where rapid and effective decision-making is crucial for responding to fluctuating conditions and maintaining system stability and efficiency.

Recognising the limitations of current computational resources in traditional approaches, integrating machine learning techniques into MPC frameworks presents a promising avenue to overcome these challenges. Specifically, this thesis focuses on supervised learning to reduce the computational burden of MPC in MLD systems. With the application of transformer models, renowned for their effectiveness in sequence modelling tasks across various fields, we may be able to learn and predict sequences in MPC problems. The main goal of the proposed approach is to learn the binary components of MILP problems, thereby reducing the computational load offline and enhancing the applicability of applying MPC in online microgrid settings while keeping optimality loss as minimal as possible.

The potential of transformers to improve the predictive accuracy and operational efficiency of learning-based MPC systems offers exciting prospects for more adaptive and resilient microgrid management. The upcoming chapters delve deeper into the specific methodologies employed, the experimental setup, and the empirical validations that underscore the effectiveness of transformers in this novel application. The goal is to demonstrate how this innovative integration can address the challenges of online MPC implementation.

Chapter 3

Methodology: Transformers for Binary Sequence Prediction

This chapter describes the methodology for developing and implementing a transformer to predict binary decisions in MILP problems related to the operation of microgrids. The goal is to reduce computation time using the transformer.

The transformer architecture employed in this study is based on the architecture originally designed for natural language processing but adapted for sequence modelling of binary decisions in microgrid operations [11]. The transformer's ability to handle sequences efficiently makes it ideal for predicting the binary sequences in MILP problems for MPC, where each decision point can significantly affect subsequent outcomes in a control sequence.

3-1 General Overview Methodology

The methodology for integrating a transformer into the MPC framework involves several key steps:

1. Data collection and preprocessing: collecting and preparing synthetic and real-world data to accurately model microgrid operations.
2. Transformer training: training the transformer to predict binary sequences representing control actions in the microgrid.
3. From predictions to control actions: converting the binary sequences predicted by the transformer into actionable control decisions within the MPC framework.

After the transformer predicts binary sequences, these sequences must be interpreted as control actions within the MPC framework. The transformation process involves converting the integer sequences produced by the trained transformer into binary decisions. Each binary

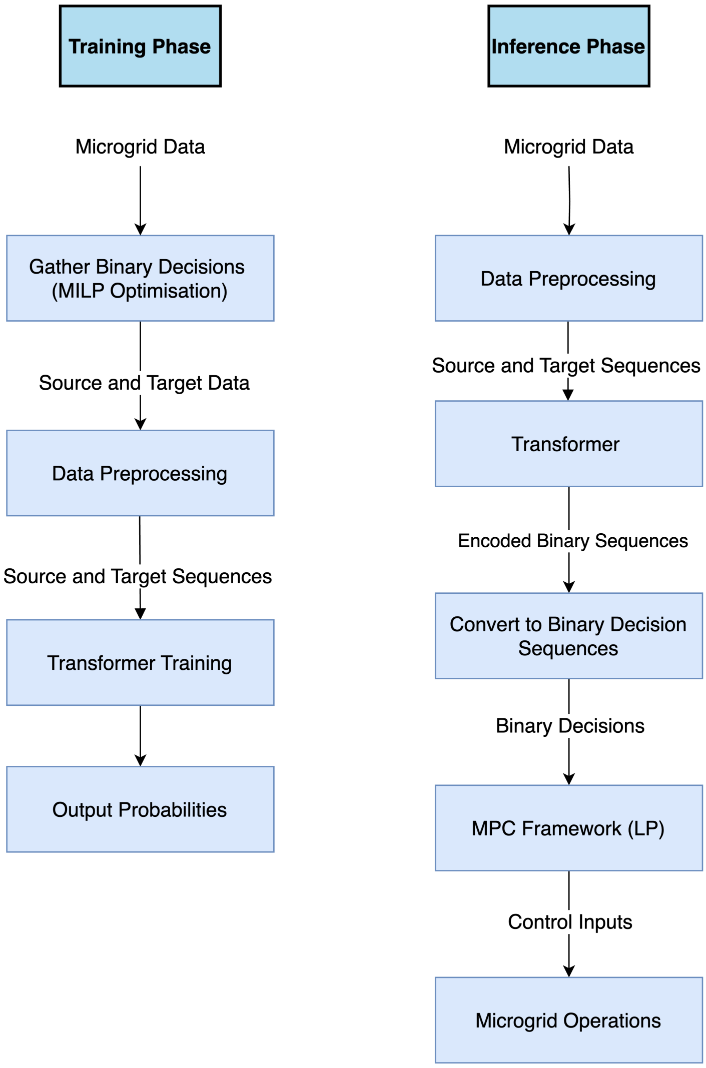

Figure 3-1: Overview methodology: training and inference Phase

value corresponds to a specific control action, such as activating or deactivating generators or exchanging with the grid. By accurately predicting these binary decisions, the transformer allows the MPC to focus solely on the continuous variables, thus converting a complex MILP problem into a simpler LP problem. Figure 3-1 overviews the training and prediction phase.

3-2 Data Collection and Preprocessing

In the context of MPC for microgrids, the inputs are essentially determined by current and future data on prices, load demands, and the system states. This dependency underscores the importance of accurate and comprehensive data collection and preparation for learning-based methods.

3-2-1 Data Collection

This research utilises a combination of synthetically generated and real-world data to effectively model microgrid operations and determine the efficacy of the learning-based approach.

Price Data Microgrid

Due to the unavailability of actual price data for energy transactions within microgrids, the price profiles — specifically the costs of producing (c_{prod}), buying (c_{buy}), and selling (c_{sell}) energy — were synthetically generated. These profiles were created using three distinct normal distributions guided by the principle that, on average, producing energy is less costly than purchasing it from the main grid and more expensive than selling it back. This synthetic generation incorporates a degree of randomness to emulate real market conditions, where the relative costs of these actions can vary. Also, the variability allows the microgrid operator to capitalise on certain market situations, such as generating and selling surplus power at a profit when production costs drop below selling prices.

Load Data Microgrid

The renewable energy generation and load profiles are based on actual data from the Dutch main grid operations, which is publicly available [32]. Given the scale discrepancy between the national grid and the microgrid scenario considered in this study, the real power data was linearly downscaled to match the capacity of our microgrid setup. This scaling adjustment ensures that the fundamental patterns and fluctuations in the energy generation and load profiles remain intact.

3-2-2 Data Preprocessing

After collection, data underwent several preprocessing steps to ensure compatibility with the transformer architecture. These steps included normalising continuous features and encoding categorical variables, preparing the data to reflect real operational scenarios and enhancing the learning efficacy of the transformer.

The transformer used in this study is a sequence-to-sequence (seq2seq) model, commonly employed in natural language processing to, for example, translate sequences, [33]. Our study processes operational data to predict binary decisions, where the source sequence includes operational parameters, and the target sequence represents binary outcomes.

Preprocessing for microgrid data involves feature embedding techniques crucial for handling inherently sequential and continuous variables like energy prices and power demands. By transforming these variables into a standardised feature dimension, the data is enriched while retaining essential information, making it suitable for the transformer framework [34].

These adaptations are vital for the transformer to effectively manage the sequence-to-sequence modelling required for MILP problems, leveraging its ability to capture temporal dependencies and relationships within the microgrid data.

Generating Training and Validation Data

To generate the training data for the transformer, the following process is used:

We start by establishing the number of MILP problems to solve, reflecting various operational scenarios over a predetermined prediction horizon. This ensures the transformer is trained on a diverse set of data, including realistic microgrid scenarios. Data collection entails acquiring real-time operational data from the microgrid, such as energy prices, power demands, and the system states, as described in the previous chapter. The optimisation problem is described in Section 2-1-2 in (2-5) and (2-7). Notably, the storage level is randomly chosen, and a random day/time is selected for the prices, load, and renewables profiles to enhance the realism and variability of each scenario. This data forms the basis of each mixed-integer linear programming (MILP) problem's constraints and conditions. Each MILP problem is solved to obtain the optimal set of binary decisions involving the control of generators, load switches, and other binary-operated components. These decisions aim for operational efficiency and cost-effectiveness. Optimal binary decisions are then extracted and paired with their respective data, forming a dataset where each entry consists of input features and the desired binary outputs.

The steps taken for data generation are as follows:

1. Formulate N_{sim} exemplary MILP problems as described in Section 2-1-2, incorporating randomly chosen storage levels and varied operational conditions based on random selections of day/time for pricing and load profiles.

2. For each problem, extract the relevant parameters:

$$Z_N = \Big\{ \Big(x_0^{(i)}, \Big\{ c_{\text{buy}}^{(i)}(k), c_{\text{sell}}^{(i)}(k), c_{\text{prod}}^{(i)}(k), \\ P_{\text{load}}^{(i)}(k), P_{\text{res}}^{(i)}(k) \Big\}_k^{N_{\text{P}}} \Big) \mid i = 1, \dots, N_{\text{sim}} \Big\}$$

3. Solve the optimisation problems using Gurobi [35].

4. Retrieve the binary decision tuples from the optimisation:

$$\Delta_N^j = \Big\{ \Big(\delta_1^{\text{on}(i)}(k), \dots, \delta_{N_{\text{gen}}}^{\text{on}(i)}(k), \delta_{\text{grid}}^{(i)}(k), \delta_{\text{b}}^{(i)}(k) \Big) \\ \mid \forall i = k + j, \forall k = 1, \dots, N_{\text{sim}}, j = 0, \dots, N_{\text{P}} - 1 \Big\}$$

With the MILP problems defined and solved, the next phase involves preprocessing the obtained data for effective learning by the transformer. This transition into preprocessing addresses the specific needs of transforming continuous and categorical variables into a structured form that supports transformer predictions.

Incorporating State Information to the Source Sequence

Most of the data we present to the transformer is already in sequential form, i.e. the price data, load data, and binary decisions. In contrast, the current state of the microgrid, which essentially describes the battery level of the system, is not sequential but remains crucial information for decision-making. We have considered two potential strategies to effectively incorporate this information into the source sequence for the transformer training.

The first strategy involves repetitively including the current state data as a vector with the length of the data sequences. This approach leverages the transformer's attention mechanism, which can discern and weigh the relevance of the current state information throughout the sequence. By presenting the state repeatedly, it is anticipated that the transformer can learn to use this information contextually at each step of the sequence prediction.

Alternatively, the current state could be included only once at the beginning of the sequence, followed by zero-padding for the remainder of the prediction horizon. Padding in transformers and other sequence models is a standard technique to equalise the lengths of sequences without introducing extraneous data, ensuring consistent processing across different inputs. This method would rely on the transformer's ability to carry forward the state information across the sequence through its internal memory mechanisms. Both methods ensure that the static state information is formatted as sequential data, accommodating the sequence-to-sequence architecture of the transformer.

For our implementation, we chose the repetitive inclusion approach, assuming that the attention mechanism within the transformer can understand the ongoing relevance of the current state across different stages of the sequence. This choice is motivated by the desire to maximise the transformer's ability to utilise available state information without assuming inherent memory capabilities and losing important information.

We have now chosen a method to transform all relevant microgrid data into sequences that are uniform in length and contain all relevant information required for the transformer prediction. Each sequence includes time-step-aligned data on energy prices, power demands, and the repeated inclusion of the current state data. This structured format ensures the transformer receives a consistent view of the microgrid's operational dynamics at each prediction step. The source sequence is presented as a tensor to the transformer as in (3-1):

$$\mathbf{S}_k = \begin{bmatrix} c_{\text{prod}}(k) & c_{\text{buy}}(k) & c_{\text{sell}}(k) & P_{\text{load}}(k) - P_{\text{res}}(k) & x_0 \\ c_{\text{prod}}(k+1) & c_{\text{buy}}(k+1) & c_{\text{sell}}(k+1) & P_{\text{load}}(k+1) - P_{\text{res}}(k+1) & x_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{\text{prod}}(N_p) & c_{\text{buy}}(N_p) & c_{\text{sell}}(N_p) & P_{\text{load}}(N_p) - P_{\text{res}}(N_p) & x_0 \end{bmatrix} \tag{3-1}$$

here $c_{\text{prod}}(k), c_{\text{buy}}(k), c_{\text{sell}}(k)$ represent the production, buying, and selling prices at time step k, respectively; $P_{\text{res}}(k) - P_{\text{load}}(k)$ captures the net power available; and x_0, the initial or current state of the microgrid, is repeated in every row to maintain its significance across the prediction horizon, N_P. This standardised sequence format is crucial for the transformer's training and inference phases, as it allows the transformer to apply its sequence-to-sequence processing capabilities effectively. The uniformity in sequence length and the inclusion of all relevant information ensure that the transformer can learn to predict the binary decisions accurately, which are critical for effective MPC in microgrids. By aligning the data in this manner, we maximise the transformer's potential to capture and utilise temporal relationships and dependencies across the sequence of inputs, thus enhancing the predictive accuracy and reliability of the MPC system. Further, by ensuring that all sequences have consistent lengths, we eliminate the need for padding tokens in the transformer, reducing its complexity and streamlining the processing.

Preprocessing Continuous Features

The continuous and discrete features are preprocessed differently to fit properly for the transformer architecture. The continuous features in the data are normalised to have a zero mean and a standard deviation of one to ensure they are on a similar scale. This normalisation is important for efficiently training neural networks. The normalisation is mathematically represented as:

$$\text{normalised\_feature} = \frac{\text{feature} - \mu}{\sigma + \epsilon} \tag{3-2}$$

where μ and σ represent the mean and standard deviation of the feature, respectively, and ϵ (a small number) ensures numerical stability by preventing division by zero. This formula is applied to continuous data before feeding them into the transformer. Normalisation can be applied across the entire dataset or within smaller batches, depending on the training strategy and variability of the data.

To handle the continuous nature of microgrid data, we introduce the *SourceFeatureEmbedder*. This component of the transformer-based architecture transforms the raw microgrid data into a format that the transformer can process effectively. Unlike traditional transformers that operate on tokenised text data, our transformer requires a nuanced understanding of real-valued input features. The *SourceFeatureEmbedder* linearly transforms these input features into a higher-dimensional space, preparing them for the subsequent layers of the transformer architecture. Further elaboration on this component is given in Section 3-3, where design choices and practical aspects of the transformer architecture are elaborated.

Preprocessing Binary Decisions

In the process of preparing data for the transformer, special attention is given to the binary decisions that represent operational commands within the microgrid, such as turning generators on or off. These decisions are critical as they directly impact the microgrid's operational effectiveness and efficiency. These sequences of binary decisions are the target sequences we provide to the transformer.

Binary decisions within the dataset consist of five binary values at each decision point, each corresponding to a specific action or state within the microgrid system. These values must be accurately processed before being inputted into the transformer. For each time step k, the binary decisions are represented as follows:

$$\boldsymbol{\delta}(k)^\top = \begin{bmatrix} \delta_b & \delta_{grid} & \delta_1^{on} & \delta_2^{on} & \delta_3^{on} \end{bmatrix}$$

where δ_b determines charging or discharging of the system, δ_{grid} the connection to the main grid, and δ_i^{on} whether the dispatchable generators are active or not.

Table 3-1: Conversion of binary representations to integer values

Binary Representation	Integer
00000	0
00001	1
00010	2
00011	3
00100	4
00101	5
⋮	⋮
11011	27
11100	28
11101	29
11110	30
11111	31

We utilised a binary encoding strategy, which manages the binary decisions efficiently. This encoding converts each time step's binary decision vector into a single integer value, creating a fixed vocabulary size. We establish $2^{n_{\text{binary}}}$ as the total number of unique combinations. This encoding reduces the dimensionality of the target sequence, transforming it into a streamlined series of integer values for the transformer. Also, employing a fixed vocabulary is desirable in standard transformer architectures as it optimises processing in the transformer architecture. Table 3-1 gives an overview of how the binary decisions are transformed into integer representations.

This approach streamlines the training process by reducing the complexity of the target data. Using a fixed vocabulary for the transformer aligns with traditional natural language processing techniques, facilitating more straightforward implementation and potentially improving the transformer's learning efficiency. Furthermore, retrieving the binary decisions from the transformer predictions is straightforward using this encoding method.

Mini Batch Processing

To optimise the transformer training, we use mini-batch data processing. Mini-batch training is a common approach in deep learning that balances computational efficiency with the ability to effectively converge on an optimal solution. It involves dividing the entire dataset into smaller subsets or "batches" that are processed sequentially during the training of a neural network. Each mini-batch consists of a specified number of samples from the dataset. The transformer's parameters are updated after processing each mini-batch rather than after the entire dataset or just one sample. This method offers a balanced approach, incorporating the advantages of both batch and stochastic gradient descent methods [34].

The choice of mini-batch processing has several advantages. First, it enhances computational efficiency because the number of simultaneously processed samples is reduced. This limits the memory overhead while leveraging the benefits of parallel processing on modern hardware. Furthermore, it helps smooth the gradient during training and often leads to better generalisation [36].

For practical implementation, the dataset is segmented into mini-batches using a DataLoader, which automates the process of sampling, shuffling, and providing batches to the transformer during training. This study chooses a mini-batch size of 32 based on empirical evidence suggesting it offers a good trade-off between training speed and system memory constraints [37].

3-3 Transformer Architecture for MILP in Microgrids

This section delves deeper into the specialised architecture of the transformer tailored for predicting binary sequences in mixed integer linear programming problems for MPC.

Source Sequence Feature Embedder

Transformers, originally designed for natural language processing, typically handle discrete input data processed as tokenised integers within a fixed vocabulary. Adapting transformers to effectively process non-discrete, continuous inputs—such as energy prices, power demands, and system states—is essential for applications like microgrid management within MPC.

Drawing inspiration from advancements in other fields, we propose the source feature embedder. Vision transformers, for instance, adapt to the continuous nature of image data by segmenting images into patches and encoding these as positions in a sequence, with each position embedded to maintain spatial relationships— a method essential for sequence recognition tasks. Similarly, we parallel this approach by embedding microgrid data linearly into a higher-dimensional vector space, thereby maintaining the dimensional consistency necessary for leveraging the transformer's powerful attention mechanisms and making the continuous data usable within the architecture [34].

The transformation applied by the source feature embedder is defined as follows:

$$\mathbf{y} = W\mathbf{x} + \mathbf{b} \tag{3-3}$$

where $\mathbf{x}$ represents the input feature vector composed of continuous data variables such as energy prices and power demands. W represents the weight matrix that projects the input features into a higher-dimensional space suitable for processing by the transformer's attention mechanisms. The bias b aids in adjusting the output vector y, aligning it precisely with the transformer model's requirements.

By transforming raw feature data into dense vectors, the source feature embedder aligns input sequences with the structural expectations of the transformer, facilitating uniform dimensionality and scale across all input features. This standardisation is crucial as it allows the transformer to apply its attention mechanisms and positional encodings uniformly across the entire sequence, enhancing the detection of subtle patterns and dependencies in the data.

Positional Encoding

In the transformer architecture, positional encoding is essential for introducing temporal context into the data. Transformers, by design, do not have any inherent mechanism to recognise the input data order. Knowing the temporal context within the data is critical for applications like microgrid control, where understanding the temporal sequence of data, such as energy prices and load demands, is crucial for effectively understanding and learning the decision-making of the MPC.

Positional encoding works by adding a vector to each input embedding. These vectors follow a specific pattern the transformer learns to associate with time steps in the data sequence. The purpose is to provide the transformer with a means of discerning the order of events within the input data, which is not naturally possible with the standard architecture due to its reliance on self-attention mechanisms that treat each input independently of its position in the sequence.

Because we preprocessed the data effectively and used the source sequence embedder, it is possible to use a relatively standard formula from [11] to apply positional encoding. Implementing positional encoding involves generating sine and cosine functions of different frequencies, where each frequency corresponds to a different position in the sequence. This can be mathematically represented as:

$$\begin{aligned} PE(pos, 2i) &= \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \\ PE(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \end{aligned} \tag{3-4}$$

where pos represents the position in the sequence, i is the dimension, and d_{model} is the total number of dimensions in the transformer (which is the same as the embedding size).

This pattern of sine and cosine functions is designed so that each position in the sequence has a unique combination of these values, which helps the transformer identify the relative or absolute positioning of the tokens in the input sequence.

3-4 Training and Prediction of the transformer

This section describes the transformer's training process and the prediction of the binary variables for the MPC controller.

3-4-1 Transformer Training

Training the transformer is a critical step in preparing it for deployment in microgrid control. This section discusses the training process, including the configuration, optimisation, and validation stages, ensuring the transformer performs optimally.

Training Setup

The training uses the cross-entropy loss function, which is appropriate for classification tasks where the output is a probability distribution across the target classes [38]. The optimisation of the transformer parameters is handled by the Adam optimiser, which is noted for its efficiency in handling sparse gradients and its adaptive learning rate capabilities.

Training involves multiple epochs where each epoch processes the entire dataset through the transformer in batches:

1. Batch processing: each batch of source data is normalised before being fed into the transformer. This normalisation standardises the data, ensuring that the transformer does not become biased or sensitive to the scale of input features.
2. Loss computation: for each batch, the transformer predicts the output and the loss is calculated by comparing these predictions against the actual targets. The loss is then back-propagated through the transformer to update the weights.
3. Optimisation Step: The optimiser updates the transformer parameters based on the computed gradients to minimise the loss function.

Several strategies ensure the transformer generalises well to new, unseen data and does not merely memorise the training set. Dropout and early stopping were successful approaches to prevent the transformer from overfitting prematurely. Integrated at various points in the transformer architecture, dropout randomly omits a subset of features in each training iteration. This method helps prevent becoming overly dependent on any single set of features and promotes the development of more robust internal representations. Further, by using early stopping, one can monitor when the performance improvement seems to halt. This indicates that the transformer is overfitting on training data and, hence, is over-training. For some measure, we can, therefore, stop the training process.

Validation and Performance Monitoring

After each training epoch, evaluation is performed using a validation dataset to monitor its performance and ability to generalise. This process mirrors the training phase but includes critical distinctions to ensure accurate performance assessment:

1. Validation loss computation: the validation loss is calculated in a manner similar to the training loss but with the transformer set to evaluation mode. This mode disables training-specific operations such as dropout and fixes parameters, ensuring no updates or learning occurs during this phase. This approach ensures that the validation performance reflects the transformer's generalisation ability rather than memorising the training data.
2. Loss analysis: validation and training losses are compared after each epoch. A stagnation of improvement in validation loss, accompanied by continued improvement in training loss, typically indicates overfitting. This signals that while the transformer is becoming increasingly accurate on training data, it is not performing better on unseen data.

Keeping track of the validation loss during training is crucial for the transformer's prediction accuracy. The validation loss gives us great insights into how well the transformer generalises on unseen data instead of overfitting.

3-4-2 Transformer Prediction

The transformer-based MPC prediction process involves two primary stages. First, the transformer generates a sequence using the trained transformer. Secondly, the predicted sequence is transformed into the binary configuration of the MPC.

Generating Predictions

The trained transformer is first set to evaluation mode to generate the predictions. In this mode, training-specific operations like dropout are deactivated. It then follows a systematic procedure for binary decision sequences in microgrid control:

1. Source feature embedding and positional encoding: Initially, the raw input data of the microgrid is processed through the source feature embedder, which adapts the continuous microgrid data into a format suitable for the transformer's sequential processing. This embedding is then enhanced with positional encoding to inject temporal context into the data, which is crucial for acknowledging the sequence order in the subsequent processing stages. This is similar to what was done in the training phase, making it consistent and understandable for the transformer architecture.

2. Encoder processing: The encoded input then traverses the transformer's encoder layers, in which the input data is refined and enriched through the different layers. The output from the encoder represents a comprehensive contextual representation of the input sequence, which sets the stage for predicting sequences accurately.

3. Sequence Generation: the sequence generation starts with an empty list which is filled iteratively:

    - Embedding and positional encoding: embeds the current sequence tokens and applies positional encoding.
    - Target mask generation: a mask is created to ensure causal decoding, preventing the transformer from accessing future sequence elements not yet generated, thus preventing future data leakage.
    - Decoder processing: processes the data through multiple decoder layers, interpreting and predicting based on the current and broader context provided by the encoder.
    - Logit calculation and token selection: generates logits from the final layer of the decoder to determine the probability distribution over possible next tokens, representing the binary configurations with the highest probability. The most probable next token is identified using $\arg\max$, appended to the sequence. This process repeats until the sequence reaches the predefined maximum length.

Using these logits from the decoder, the transformer identifies the most probable next token, appending it to the ongoing sequence. This loop repeats until the sequence is fully generated, adhering to the predefined maximum length to ensure completeness and relevance of the output.

Table 3-2: Conversion of integer values to 5-bit binary representations

Integer	Binary Representation
0	00000
1	00001
2	00010
3	00011
4	00100
5	00101
⋮ ⋮	⋮ ⋮
27	11011
28	11100
29	11101
30	11110
31	11111

The result is a complete sequence of tokens representing the transformer's prediction for the binary decision-making process, ready to be transformed into actionable binary configurations for microgrid control.

Transform Predictions in binary configurations

To enable the practical application of the MPC, the integer values generated by the transformer must be converted back into a format that suits within the MPC framework. As outlined in Section 3-2, binary decision representations within the microgrid are initially represented as integers to simplify the data for the transformer during training. These integers are then converted back to their corresponding 5-bit binary representations, which directly map to operational commands such as turning generators on or off.

Table 3-2 illustrates this conversion process from integers to binary decisions. Appendix A shows a complete conversion chart. This step is essential for translating the transformer's predictions into the binary sequences necessary for simplifying MILP problems into LP problems within the hybrid MPC framework.

The entire integration of the transformer into the MPC for microgrid operations is discussed in Section 3-5.

3-5 Integration with MPC

This section details the integration of the transformer-based binary decision predictions into the MPC to simplify MILP problems into more manageable LP problems, enhancing the online applicability of the MPC in microgrids.

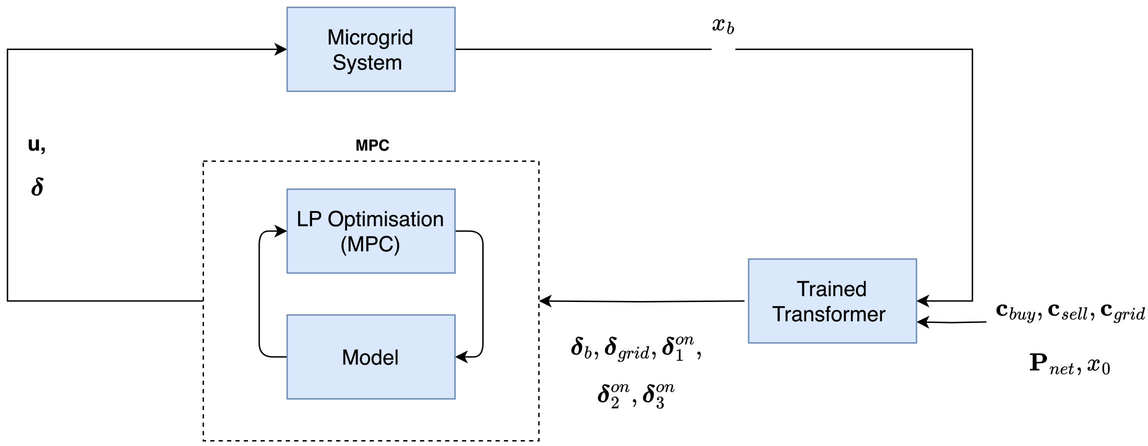

Figure 3-2: Transformer-based MPC

Closed-Loop MPC Implementation

The integration begins with the transformer's ability to predict binary sequences from the source inputs: pricing data, predicted load demands, and the microgrid's current state to the trained transformer. These predictions are then seamlessly fed into the MPC controller.

Upon receiving these binary sequences, the MPC incorporates them as fixed parameters within its optimisation framework. The model predictive control (MPC) problem is described in Section 2-1-2 in (2-5) and (2-7). Fixing these binary values reduces the problem to an linear programming (LP) problem. This is a crucial shift from the typical MILP problem-solving process to a simplified LP approach. By treating the binary decisions as known quantities, the MPC can now only focus on optimising the continuous variables of the microgrid's operation. This simplification removes the computational burden typically associated with binary variables in MILP, allowing for faster decision-making. Once the LP problem is solved, the MPC controller applies the first set of inputs of the computed control sequence to the system based on the optimised outcomes.

Following the application of the control signal, the microgrid system's state is updated, incorporating changes in energy prices and load demands. This updated information feeds into the transformer, predicting new binary sequences for the next cycle. The MPC then recalculates the control actions based on these new predictions, completing the loop. A schematic overview of the MPC control scheme is given in Figure Figure 3-2.

Through this integration, the transformer and MPC enhance microgrids' operational efficiency, streamlining the decision-making process and adapting in real-time to the evolving energy demands and operational conditions. The upcoming section details how the performance of the transformer-based MPC can be evaluated and how well the transformer-based MPC performs compared to the optimal controller.

3-6 Evaluation Metrics and Validation

This section outlines the methods used to evaluate and validate the performance of the transformer-based MPC compared to traditional and learning-based MPC approaches.

Loss Metrics During Training

To ensure effective and stable training, the transformer training is monitored by comparing training loss against validation loss after each epoch. The data is partitioned into an $80:20$ split for training and validation. The training loss assesses the transformer's fit to the training data, while the validation loss gives insight into generalisation by evaluating performance on a separate set. Frequent monitoring helps adjust the hyperparameters effectively. Signs of overfitting are indicated when validation loss stagnates or increases while training loss decreases, indicating the need for adjustments in training strategy, such as modifying learning rates or early stopping.

Optimality Gap Measurement

When the transformer is trained, the transformer's performance is further evaluated through the optimality gap metric. This involves solving the MILP using a standard solver like Gurobi to obtain an optimal objective value, which is then compared against the value achieved using the transformer-predicted binary sequences. Calculating this gap across numerous simulations provides insights into the accuracy of the transformer's predictions and its consistency in yielding feasible solutions. A low optimality gap signifies high predictive accuracy. The optimality gap is computed using the following:

$$optimality\_gap = \frac{c_{\text{tf}} - c_{\text{opt}}}{c_{\text{opt}}} \tag{3-5}$$

where c_{tf} is the average cost of the transformer and c_{opt} the average cost of the optimal controller.

MPC Operation

Operational efficiency is tested by implementing the transformer-based MPC in a simulated microgrid environment, as described in section 3-5. The simulation tests the transformer-based approach in realistic operational scenarios to evaluate its applicability and flexibility. This simulation helps determine how well the transformer-based MPC adapts to changes and manages grid operations, providing a practical measure of its utility in dynamic conditions.

Benchmark LSTM

A comparative analysis is conducted against a benchmark LSTM-based MPC, established under similar training conditions to ensure a fair comparison. This is particularly significant as LSTM is also a sequence-modelling technique, allowing for similar data preprocessing methods and the same dataset. This approach ensures a direct and fair performance contrast between the transformer and an established learning-based approach, as highlighted in [18]. Performance metrics such as computation time, prediction accuracy, and feasibility rates comprehensively evaluate each approach's strengths and weaknesses in MPC applications.

In Chapter 4, the results of simulations and experiments are shown and evaluated elaborately. This chapter shows how well the described transformer works in practice and aims to unveil the strengths and weaknesses of applying transformers in MPC for microgrids as described in this chapter.

3-7 Conclusion

This chapter has detailed developing and implementing a transformer tailored for predicting binary decision sequences within MILP formulations for microgrids. The transformer, traditionally used in natural language processing, aims to enhance the computational efficiency of MPC systems by simplifying the decision-making process. By effectively translating predicted binary sequences into actionable control decisions within the MPC framework, the transformer facilitates a transition from MILP to simpler LP problems.

The transformer has been specifically adapted to process microgrid operational data, integrating continuous and discrete data types into one learning framework. This adaptation includes developing data preprocessing techniques that convert operational parameters into a sequence format conducive to transformers. Central to these innovations is the source feature embedder, which effectively handles continuous inputs such as energy prices, power demands, and system states. Inspired by advancements in fields like vision transformers and other machine learning applications, this component linearly transforms these inputs into a higher-dimensional vector space, aligning them with the transformer's architecture to enhance its capability to leverage powerful attention mechanisms. This ensures dimensional consistency and facilitates the detection of subtle patterns and dependencies critical for microgrid decision-making, mirroring techniques used across different technological domains to handle continuous data effectively.

Further preprocessing of the microgrid data involved incorporating the microgrid's current state information into the sequence prediction was introduced. By repetitively including state information rather than padding, we assumed the transformer's ability to utilise its attention mechanisms dynamically, thereby improving prediction accuracy and responsiveness to state changes in MPC operations.

We established a method for generating training and validation data, which is crucial for the tr transformer's ability to learn and generalise across varied microgrid operations. Enhanced by mini-batch processing, this methodology is expected to improve computational efficiency and strengthen the model's ability to generalise, which is crucial for its application in real-world microgrid management.

We proposed measures to validate and measure the efficacy of the transformer-based MPC approach. The following Chapter Chapter 4 delves into detailed simulations and empirical validations of the transformer. These simulations showcase the practical applications of the methodologies developed in this chapter and evaluate the transformer's performance in various scenarios.

Chapter 4

Results and Simulations: Transformer-Based MPC for Microgrids

This chapter presents the results and analyses from the simulations conducted to evaluate the performance of the transformer developed for predicting binary decisions in MPC of microgrids. The implementation aims to address the computational challenges associated with MILP by leveraging the predictive power of transformers.

Following a detailed description of the experimental setup, this chapter outlines the process and outcomes of various simulation scenarios designed to test the transformer's efficacy under different operational conditions. These simulations assess its ability to accurately forecast binary decisions and its impact on the optimisation and efficiency of model predictive control (MPC) in microgrids. Key aspects of this evaluation include the analysis of training and validation performance, optimality gap measurement, and reliability in finding feasible solutions. Additionally, the chapter compares the prediction time of the transformer to benchmarks, which gives insightful data on its practical viability and potential integration into existing microgrid control systems.

Through testing and validation, we aim to demonstrate the capabilities of the transformer and its practical implications in improving the computational efficiency of MPC.

4-1 Overview of Experimental Setup

This section details the experimental setup used to evaluate the transformer designed for MPC in microgrids. The experiments are designed to replicate real-world operational scenarios that microgrids encounter, aiming to test the approach's reliability, speed, and accuracy under diverse conditions. We also discuss the choice of hardware and software, which is crucial for meeting the demands of advanced machine learning algorithms.

4-1-1 Simulation Environment

The simulations were conducted on a MacBook Pro equipped with an M1 Pro chip, utilising Python and PyTorch as the machine learning library. While the M1 Pro includes an integrated GPU, it does not support CUDA, a critical framework for accelerating computing operations on NVIDIA GPUs, widely recognised for enhancing transformer computations. Due to the absence of CUDA, the experiments in this chapter were conducted on the CPU, which is less optimal for the parallel processing demands of transformers. This limitation contributed to longer than desirable inference times. To address this, we performed an experiment that optimised prediction time with the available resources by implementing the transformer in ONNX runtime. ONNX allows for creating serialisable and optimisable models from PyTorch code, and by transitioning to ONNX, the transformer is compiled into a graph representation that runs more efficiently at Runtime. This optimisation proved crucial for deploying machine learning models in production, where execution speed is as important as accuracy. Further details of this experiment are given in Appendix Section B. This step is part of ongoing efforts to refine the transformer approach's applicability and effectiveness in operational settings, ensuring that the transformer predicts accurately and responds swiftly to dynamic operational conditions.

4-1-2 System Model and Assumptions

This section establishes the theoretical and operational framework used throughout our experiments. As detailed in Chapter 2 and further elaborated in Appendix C, the system model provides the foundational MLD formulation upon which our transformer operates. Understanding this model is essential for interpreting the training results and their applicability to real-world scenarios. The key assumptions that hold for the simulations chapter are given below:

- Data availability: we assume complete knowledge of predicted price and load data for the microgrid system over the prediction horizon. This assumption allows us to focus on the accuracy of binary decision predictions without the added variability of uncertain input data. The details of how the data is gathered are elaborated in Chapter 3.

- Power generated by renewable energy Sources: we assume that the power generated by renewable energy sources within the microgrid system is known. Although this is not precisely known in practice, this would significantly complicate the MPC problem. This assumption simplifies the complexities we need to manage in our simulations. However, it's important to note that this assumption is not fundamental to the applicability of our approach. The methodologies and strategies we discuss could also be used in scenarios where only predicted prices, loads, and renewable generation values are available, which is more reflective of real-world conditions where exact future generations cannot be precisely predicted.

- Consistent system configuration: a fixed number of generators are assumed, simplifying the control problem's complexity and ensuring that the training data reflects consistent operational conditions.

These assumptions are pivotal for structuring our experiments and are designed to isolate the performance in controlled conditions. By maintaining a consistent system model and set assumptions, we can accurately assess the impact of different hyperparameter configurations on the transformer's ability to learn and generalise effectively.

4-2 Training Results

With the system model and foundational assumptions laid out, this section delves into the outcomes of the training phase for the transformer. We explore configurations and their effects on the transformer's training time and loss computation in a controlled simulation environment.

4-2-1 Hyperparameter Configuration and Training Process

The development of the transformer begins with careful tuning and setting up of hyperparameters, crucial for optimising the learning dynamics specific to microgrid control applications:

- Vocabulary sizes: we use a target vocabulary size of 32, reflecting the transformer's capability to predict 32 unique binary outcomes. This parameter is critical because it specifies the possibilities in the output space that the transformer can predict. The source vocabulary does not have a specified size due to the continuous nature of the input data.
- Architectural configuration:
  - Model dimensionality: Refers to the size of the transformer's hidden layers and embeddings. This affects the transformer's overall architectural configuration and impacts each part of the architecture's processing pipeline, from input handling to producing final outputs. It is a core attribute that determines the capacity of the transformer to process and represent data, influencing both the computational power required and the depth of data representation achievable.
  - Number of heads: this parameter controls how many different parts of the sequence are attended to independently in multi-head attention mechanisms. More heads allow the transformer to capture a wider range of dependencies but increase computational complexity.
  - Number of layers: each layer allows the transformer to learn another level of abstraction in the data. More layers typically enable deeper understanding but also increase the risk of overfitting and the computational burden.
  - Dimension of feedforward network: this defines the size of the feedforward layers within each transformer block. A larger dimension can enhance the transformer's capability to understand complex relationships but may lead to overfitting and extended training durations.
- Sequence handling and regularisation:

  - Maximum sequence length: corresponds to the fixed prediction horizon length, with longer sequences increasing task complexity and computational demand as the transformer's attention mechanism needs to span wider data stretches.
  - Dropout: a regularisation method to prevent the transformer from overfitting by not overly depending on any single feature. Higher dropout rates can aid generalisation, though excessive rates might impede learning

- Optimisation parameters:
  - Learning rate: dictates the speed at which the transformer updates its weights during training. A lower rate might slow down learning but can lead to more stable convergence.
  - Number of epochs: represents how often the training algorithm works through the entire training dataset. More epochs allow for more thorough learning but can lead to overfitting without adequate stopping criteria.

We began the training process by collecting data as outlined in Section 3-2-2. After solving 16,000 mixed-integer linear programming (MILP) problems, we partitioned the data into a training and validation set in an 80-20% ratio. This ensures sufficient data volume for training and reliable validation, aligning with other studies performing similar tasks, e.g. in [10] and [18].

The transformer was trained using a batch-based approach, normalising each batch using global statistics from the training set to maintain consistent input feature scaling. Adjustments to hyperparameters, including changes to model dimensionality, number of heads, dropout rates, and learning parameters, were evaluated to discern their trade-offs between enhancing the transformer's representational capacity and the required computational time. This calibration aimed to find a good balance, ensuring that the transformer achieves high precision in predicting binary sequences for microgrid control and maintains computational efficiency, which is crucial for real-time applications

Training Outcomes

After adjusting and testing various hyperparameters, we can distinguish two approaches for configuring the transformer. First, we can prioritise higher complexity with increased model dimensionality, number of heads, and layers. The other takes a more minimalist approach with fewer layers and reduced complexity, aiming for faster training times but potentially less accuracy. We also performed experiments with two different sequence lengths to see the effect of longer sequences and whether the transformer still performs well for a longer sequence length.

High Complexity Configuration

The first configuration includes high dimensionality, multiple attention heads, and several layers. This setup is expected to capture intricate dependencies within the data, thereby improving the transformer's ability to generalise and make accurate predictions in new, unseen scenarios. However, this high complexity carries the risk of overfitting, and it can be

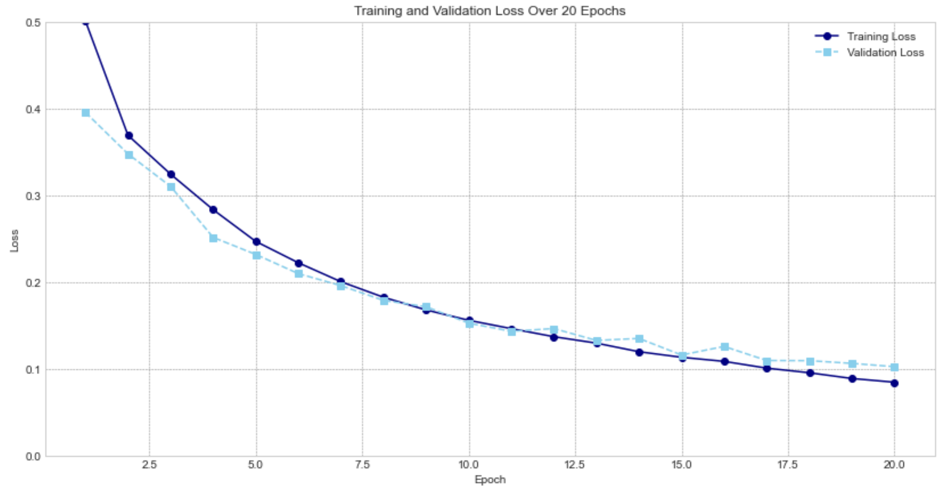

Figure 4-1: Training and validation loss of higher complexity configuration, total training time: 158.4 minutes

expected that inference will take longer with a more complex configuration. This complexity is reflected in the substantial number of trainable parameters— precisely, 23,168,544. This large scale of parameters can significantly impact computational load, potentially leading to longer inference times, which are less desirable for real-time applications. Despite these concerns, it is beneficial to compare the higher complexity to the lower complexity configuration to see how prediction accuracy is affected by the configuration of hyperparameters.

The hyperparameter configuration is depicted in Table 4-1. Figure 4-1 shows the training and validation results of the higher complexity transformer. It can be seen that the training and validation loss initially decrease, which implies that the transformer learns and generalises well. However, after 20 epochs, the validation loss is platooning while the training loss is still decreasing. This indicates that the transformer is starting to overfit the training data, so the training has been stopped. The training time of this configuration is 158.4 minutes. This is significantly higher than when compared to a transformer of lower complexity. However, we assume that the transformer may learn more intricate dynamics in the data since the losses are lower than the final loss in the other configuration. This makes the training time possibly worthwhile.

Low Complexity Configuration

The second configuration uses fewer layers and a reduced number of dimensions and heads. Although it is expected that this will significantly decrease training time, it is possible that the transformer will not be able to learn the most intricate dependencies in the data and, therefore, generalise in new unseen scenarios. However, using a less complex configuration may have several advantages. First, the training time will be significantly reduced due to the lower complexity, primarily because the number of trainable parameters is lower, specifically, 1,261,344 parameters. This reduction in complexity speeds up training and benefits inference performance. We use the trained transformer to predict sequences for the MPC in the inference phase. A lower complexity means that the source sequence passes through a less

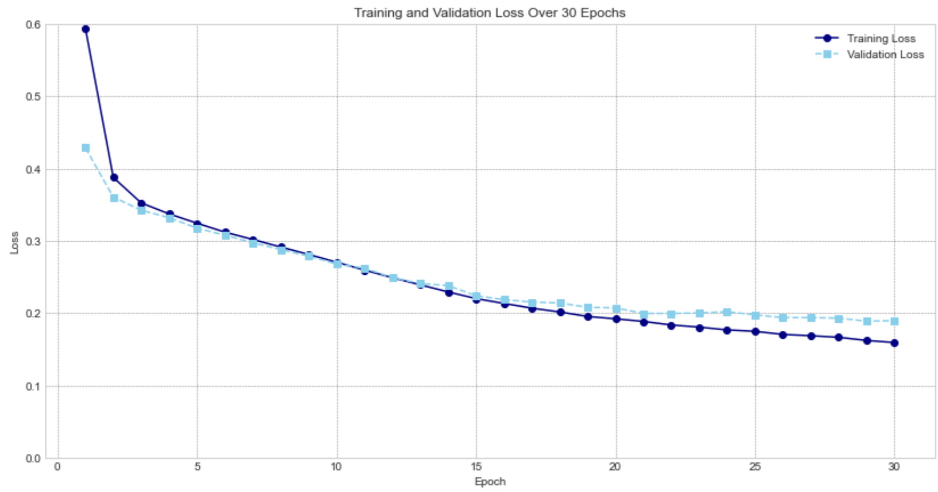

Figure 4-2: Training and validation loss of lower complexity configuration, total training time: 14.9 minutes

complex transformer, which speeds up the prediction time—a critical advantage in the online control of the MPC.

Figure 4-2 shows a training and validation loss plot for the lower complexity trained transformer. The plot illustrates an initial significant decrease in training and validation losses, indicating effective learning and generalisation. However, the validation loss platoons as training progresses, suggesting the start of overfitting. This configuration significantly reduced training time to 14.9 minutes, which is beneficial for scenarios requiring quick deployment or limited computational resources. While the reduced complexity accelerates training and inference, potentially enhancing real-time applications like MPC, it may struggle with complex data dependencies, highlighting a trade-off between accuracy and inference time. This underscores the need to balance complexity against specific application demands and constraints. Further investigation into this trade-off must be done before conclusions can be drawn on prediction accuracy.

Preliminary Conclusions Training Phase

The training results indicate a strong start with relatively low initial loss values and a steep decline, suggesting that the transformer can quickly learn the underlying patterns in the microgrid data. However, the tendency towards overfitting, as indicated by the platooning of validation losses, suggests that while the transformers capture the training data well, they may not generalise as effectively to new, unseen data without further adjustments to prevent overfitting. Further, we see that the number of epochs needed for training is relatively low compared to other learning-based methods. Results over extended training periods showed that prolonged training did not yield improvements, as the validation loss consistently platooned or even increased, indicating primarily overfitting rather than learning to generalise from new, unseen data.

High and low-complexity configurations demonstrate the transformer's adaptability to different operational demands. While potentially offering better generalisation in complex sce-

Table 4-1: Transformer configuration parameters

Parameter	**Low complexity**	**High complexity**
Source Vocabulary Size	-	-
Target Vocabulary Size	32	32
Model Dimension (d_{model})	128	256
Number of Heads	8	16
Number of Layers	1	8
Dimension of Feedforward Network (d_{ff})	2048	2048
Feature Dimension	5	5
Maximum Sequence Length	N_P	N_P
Dropout	0.1	0.2
Number of Epochs	30	20
Learning Rate	0.0001	0.0001
Training Time	14.9 minutes	158.4 minutes
Nr. Trainable Par.	1,261,344	23,168,54

narios, the higher complexity configuration requires careful management due to its longer training times and a greater risk for overfitting. On the other hand, the lower complexity configuration promises quicker training and inference times, making it more suitable for online applications, though possibly at the expense of capturing more complex data relationships.

These findings will guide future iterations of tuning, particularly in exploring combined approaches that balance complexity with computational efficiency. The subsequent sections will delve deeper into how these configurations affect prediction accuracy and the practical deployment of the transformer in microgrid management.

4-3 Prediction Accuracy of Transformer-Based MPC in Open Loop

With multiple transformers trained under varying configurations, this section evaluates their prediction accuracy. Critical to our analysis is the optimality gap, a key measure that quantifies the performance of the transformer. The optimality gap measures the percentage difference between the optimal solution and the solution derived from the transformer's binary predictions for different MILP parameterisations.

Alongside the optimality gap, we also use the success rate of the transformer as an important metric. Defined as the percentage of instances where the transformer produces feasible solutions in MILP, this metric gives insight into the reliability and practical utility of the novel approach. Another essential aspect of the evaluation is the solution time, which benchmarks the speed of our transformer-based approach against conventional solvers. This section will help understand transformer-based MPC 's real-world applicability and effectiveness in controlling microgrids. This section evaluates the performance in open-loop. Feedback is included in Section 4-4.

4-3-1 Experimental Setup

We initially examine the accuracy of predictions in open-loop scenarios to evaluate the effectiveness and reliability of the transformer-based MPC for microgrids. Given that MPC within mixed logical dynamical (MLD) systems fundamentally involves solving MILP problems to determine optimal control strategies—which this study aims to simplify—our focus here is primarily on this aspect of control. We redefine the MILP used in this simulations and refer to Section 2-1-1 and Appendix C for further details:

$$
\begin{aligned}
&\min_{\boldsymbol{P}_{\text{dis}}(k), P_{\text{grid}}(k), P_{\text{b}}(k), \boldsymbol{\delta}(k), \boldsymbol{z}(k)} J(\boldsymbol{P}_{\text{dis}}(k), C_{\text{grid}}(k), c_{\text{prod}}(k)) \\
&\text{subject to:} \\
&\quad E_1\boldsymbol{\delta}(k) + E_2\boldsymbol{z}(k) \leq E_3\boldsymbol{u}(k) + E_4, \\
&\quad P_{\text{b}}(k) = \sum_{i=1}^{N_{\text{gen}}} P_i^{\text{dis}}(k) + P_{\text{res}}(k) + P_{\text{grid}}(k) - P_{\text{load}}(k), \\
&\quad \underline{P}_{\text{b}} \leq P_{\text{b}}(k) \leq \bar{P}_{\text{b}}, \\
&\quad \underline{P}_{\text{grid}} \leq P_{\text{grid}}(k) \leq \bar{P}_{\text{grid}}, \\
&\quad \delta_i^{on}\underline{P}_{\text{grid}} \leq P_{\text{grid}}(k) \leq \delta_i^{on}\bar{P}_{\text{grid}}, \\
&\quad \underline{x}_{\text{b}} \leq x_{\text{b}}(k) \leq \bar{x}_{\text{b}}, \\
&\quad \boldsymbol{u}(k) = [\boldsymbol{P}_{\text{dis}}^{\top}(k), C_{\text{grid}}(k), P_{\text{b}}(k)]^{\top}, \\
&\quad \text{for } i = 1, \ldots, N_{\text{gen}}, \\
&\quad \text{for } k = 0, \ldots, N_{\text{p}} - 1.
\end{aligned}
\tag{4-1}
$$

Accordingly, we assess how accurately the transformer-based approach predicts outcomes and its ability to find feasible solutions consistently. The experiments include solving (4-1) for various realistic states, pricing, and load conditions typical of microgrids. By conducting a set of these MILPs, we seek to cover a wide range of potential operational challenges the system might encounter.

Each MILP problem is first solved using the commercial solver Gurobi to establish a benchmark for the optimal solution [35]. These solutions serve as a baseline against which the solutions utilising the transformer predictions are compared. Subsequently, the same MILPs are processed using the transformer to predict the binary decisions, $\boldsymbol{\delta}$, in (4-1), which simplifies the problems into linear programming (LP) problems. This reduces their computational complexity and potentially speeds up solution times. The primary measures for evaluation include the optimality gap, the success rate and the solution time.

This experimental framework tests the transformer's predictive accuracy and highlights its efficiency and reliability in practical conditions. The outcomes of these experiments are expected to provide insights into the capabilities of transformer-based MPC in microgrid settings. In the following sections, we present the results of these experiments, analysing them in the context of the established metrics and discuss their implications for future applications in microgrid management.

4-3-2 Performance Evaluation of Transformer-Based MPC

This section delves into the evaluation of the transformer-based MPC. The single transformer refers to the initial approach where one trained transformer was used to make all predictive decisions. Subsequently, we analyse the cascaded transformer, which uses multiple transformers in transformer-based MPC.

Performance Evaluation with Single Transformer

The system's initial configuration employed a single transformer, assuming this would be sufficient for managing the diverse and dynamic conditions typical of microgrid operations. This section explores the outcomes and challenges encountered with this setup. Several hyper-parameter configurations have been tested to evaluate the single transformers. Further, the transformers have been assessed on two different horizon lengths, i.e. 25 and 48, representing 12h and 24h, respectively.

To evaluate the performance of our single transformer, we solve 1000 open-loop problems from the validation dataset representing different microgrid scenarios. The core of our evaluation involves two key steps: first, solving the MILP problem to optimality to establish a benchmark for optimal outcomes; second, assessing the transformer's predictions and solving the problem using the transformer's predictions. We measure performance primarily through three metrics: the success rate, the optimality gap, and the average solution time. These metrics directly indicate the transformer approach's reliability and precision in a practical setting.

Table 4-2 and Table 4-3 show the results of several experiments run using different hyperparameter configurations for two prediction horizon lengths. From Table 4-2, it can be deduced that the transformer can give accurate predictions for the optimisation during the majority of the time for $N_{\mathrm{P}} = 25$. The optimality gap is consistently low, while the success rate is also higher than 90%. However, the optimisation time required for the transformer-based approach is higher than the time it takes to use a commercial solver. This result was not initially expected since we have used the trained transformer to simplify the problem from a MILP problem to an LP problem. This indicates that the time required for the transformer-based predictions exceeds that of solving the problem directly with a commercial solver. This is substantiated further when we solely analyse the Runtime of the LP problem. This Runtime is only 0.00063s, which means that predicting the binary values for the optimisation problem is the bottleneck.

The prediction time problem is especially visible when we use a more complex configuration, which increases the inference time even further. The prolonged inference time can be attributed to the higher complexity of the transformer. Transformers, by design, are complex constructs with potentially millions of parameters, necessitating a substantial number of computations, especially during the sequence-building phase of prediction. Each prediction requires the transformer to process through its extensive network to generate the sequence iteratively. This significantly impacts the overall efficiency of the solution process. Further optimisation of the transformer is necessary to address the inefficiencies in inference time. Appendix B further examines an exemplary approach.

When we analyse Table 4-3, we observe that the transformer struggles to achieve similar performance over longer horizons ($N_{\mathrm{P}} = 48$). First, it can be seen that the success rate

Table 4-2: Performance evaluation of transformer for different configurations, Horizon = 25

Model	**Success Rate [%)]**	**Optimality Gap [%]**	**Avg. Time [s]**	**Gurobi [s]**
Low complexity	92.30	1.09	0.0469	0.0039
High complexity	76.80	0.66	0.3272	0.0039

Table 4-3: Performance evaluation of transformer for different configurations, Horizon = 48

Model	**Success Rate [%)]**	**Optimality Gap [%]**	**Avg. Time [s]**	**Gurobi [s]**
Low complexity	23.5	1.14	0.1141	0.0068
High complexity	61.30	0.59	0.4921	0.0068

has dropped significantly, especially for those with lower complexity. This indicates that the transformer of low complexity cannot accurately generalise to achieve accurate predictions over the majority of time. However, if the transformer predicts a feasible solution, it predicts with only a slight optimality gap. We can see that the success rate and optimality gap improved when using the higher complexity, which comes at the cost of severe prediction times. This indicates that further tuning is required alongside computational optimisation of the transformer to achieve better and online applicable results over longer horizons.

From the experiments over longer and shorter horizons, it can be deduced that while the transformer generally provides accurate predictions for optimisation, achieving consistently low optimality gaps, the success rate is disappointingly lower than anticipated. The success rate is relatively low in too many cases, indicating that the transformer often fails to produce feasible optimisation solutions. Given the undesirably low success rate, we implement a fallback strategy involving multiple transformers, as will be discussed further in the upcoming section. This strategy employs several transformers trained under slightly varying conditions to provide backup predictions when the primary transformer fails to yield a feasible outcome. This approach aims to enhance the reliability of the MPC system.

Performance Evaluation of Cascaded Transformers

Given the challenges encountered with the single transformer configuration—particularly its relatively low success rate in producing feasible solutions—a more robust approach is assumed to be necessary. Hence, we have implemented a cascaded fallback strategy involving multiple transformers to enhance the reliability of the predictive control system. This section explores the performance of this modified system architecture, where multiple transformers, each trained using slightly different hyperparameter configurations, are employed sequentially. The single transformer evaluation concluded that the shorter horizon could use lower complexity configurations while the longer prediction horizon required higher complexity configurations. This cascade setup is designed to provide backup predictions whenever the primary transformer fails to deliver a feasible solution, thereby increasing the overall success rate of the system.

The decision to adopt a cascaded approach stems from initial observations that, while the transformer generally delivered accurate predictions for optimisation with consistently low optimality gaps, the overall success rate is insufficient for effective MPC operation; we would require fallback strategies too often when only using one transformer. This discrepancy was mainly observed when dealing with more complex configurations and longer prediction horizons, where the single transformer often struggled to maintain performance consistency.

In this fallback strategy, control is passed to the next transformer in the cascade if the primary transformer fails to predict a feasible solution. Each subsequent transformer in the sequence offers a slightly different interpretative perspective on the input data, potentially capturing nuances that the previous one missed. This layered approach increases the likelihood of obtaining a feasible solution and enhances the system's robustness against the diverse operational challenges encountered in microgrid management. We chose a configuration with a high success rate as the first transformer in the cascade and hope the other transformers will increase the success rate.

While we anticipate that the cascaded approach will improve the success rate of feasible solutions, it comes with the inherent drawback of potentially requiring more predictions in some scenarios. Each additional prediction step adds to the total computation time, increasing the overall solution time. This trade-off between increased success rates and prolonged solution times is a critical factor that must be considered when evaluating the effectiveness and efficiency of the cascaded transformers.

The performance of these cascaded transformers is assessed using the same measures previously applied: success rate, optimality gap, and average solution time. Special attention is given to comparing the success rates before and after implementing the fallback strategy to quantify the improvements this approach brought about explicitly.

The forthcoming analyses and results will detail the effects of additional transformer layers on the MPC's ability to handle increasingly complex scenarios without sacrificing operational efficiency or decision-making speed. These insights are crucial for validating the fallback strategy's effectiveness and for informing future enhancements to the transformer-based MPC framework. Algorithm 1 shows the cascaded transformer prediction approach.

Results of Cascaded Transformers

Implementing a cascaded strategy involving multiple transformers shows a clear improvement in the success rates of feasible solutions as more transformers are added to the system. The plot below (Figure 4-3) and the accompanying table (Table 4-4) illustrate how the success rates increase with each additional transformer in the cascade.

As shown in the figure, the success rate initially starts at 89.5% with a single transformer and increases to 98.10% with four transformers. This significant improvement underscores the benefit of the fallback strategy, where each subsequent transformer in the cascade can potentially correct the shortcomings of its predecessors. The optimality gap remains low across different configurations.

We see similar results for the longer horizon open-loop MPC, where $N_{\mathrm{P}} = 48$. Again, the success rate increases by including more transformers. However, the final success rate is significantly lower than the shorter horizon's success rate, as seen in Table 4-5. Also, the average solution time for this application has increased. This is due to multiple factors: first, since the success rate is lower overall, we need to predict for all four transformers more often than the other experiment. Second, since the transformer predicts the sequences iteratively,

Algorithm 1 Cascaded transformer prediction

```
Input: microgrid_sequences, transformer_list[]
Output: feasible_solution, total_inference_time
Initialise:
  feasible_solution ← NULL
  total_inference_time ← 0
  model_count ← 0
for each model in transformer_list do
    Start timer
    prediction ← model.predict(input_sequence)
    Stop timer
    total_inference_time += timer.duration
    model_count += 1
    optimisation_result ← run_optimisation(prediction)
    if optimisation is feasible then
        feasible_solution ← optimisation_result
        break
    end if
end for
if feasible_solution is not NULL then
    return feasible_solution, total_inference_time
else
    Log: "No feasible solution found after {model_count} models."
    return NULL, total_inference_time
end if
```

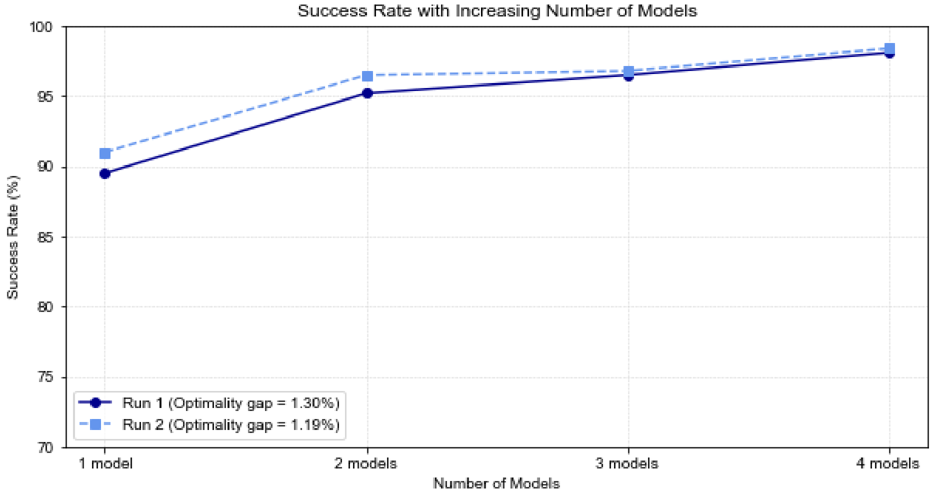

Figure 4-3: Success rate with increasing number of transformers for $N_\mathrm{P} = 25$

Table 4-4: Success rates for cascaded transformers, $N_\mathrm{P} = 25$

Number of Transformers	Success Rate Run 1 (%)	Success Rate Run 2 (%)
1 transformer	89.50	91.00
2 transformers	95.20	96.50
3 transformers	96.50	96.80
4 transformers	98.10	98.40
Optimality gap	1.30%	1.19%

Average MILP time: 0.0040

Average LP time: 0.0729 seconds

a longer sequence will logically lead to longer prediction times and, thus, longer optimisation times. What should be noted is that the average optimality gap of the longer horizon transformer-based MPC is very low, which indicates that when a solution is found for the transformer, the prediction is very accurate. This is also because we needed to transformers of higher complexity compared to shorter horizons to achieve good results. We saw in the previous section that this can reduce the optimality gap.

Especially the success rate is concerning, as this is significantly lower than for the shorter horizon MPC. The lower success rate for the longer horizon MPC suggests limitations in the transformer's scalability and ability to handle complex, extended sequences. This decrease in success rate may stem from several specific challenges. First, the training dataset may not adequately capture the complexities of longer-duration dynamics, lacking sufficient scenarios that mimic extended operational sequences encountered in real-world settings. We used the same amount of data for both horizons while increasing the horizon might require more data. Future enhancements could include diversifying the training scenarios to better encompass these extended periods by increasing the amount of training data to improve outcomes. Moreover, further tuning of the transformers to find a more optimal configuration of the transformers for longer horizons would be beneficial.

Table 4-5: Success rates for cascaded transformers, $N_{\mathrm{P}} = 48$

Number of transformers	Success Rate Run 1 [%]
1 transformer	58.50
2 transformers	69.20
3 transformers	74.30
4 transformers	76.00
Optimality gap	0.50%

Average MILP time: 0.0068

Average LP time: 0.5044 seconds

In the shown experiments, the number of transformers was not increased further than four. Other experiments on expanding the cascaded transformers have indicated diminishing returns in success rate improvements with adding more transformers. While the number of transformers initially significantly boosted the success rates, subsequent additions have shown only marginal gains. These minimal improvements do not compensate for the potential increase in computation time required, suggesting a point of diminishing returns in the cascading strategy. Given these findings, alternative approaches must be considered for scenarios requiring a feasible solution, such as closed-loop MPC. We discuss two potential strategies.

When the transformer fails to deliver a feasible solution, reverting to solving the MILP directly with a commercial solver might be advisable despite the high computational demand. This approach guarantees a solution but at a higher computational cost. The direct solution of MILPs should be reserved for exceptional cases where the transformer-based approach fails to meet the required thresholds of feasibility and efficiency. From Figure 4-3, this will only be the case in less than 2% of the time.

Another viable strategy involves implementing a simple, rule-based fallback system. This system would activate when predictions fail to achieve a solution quickly. Rule-based systems, while less sophisticated and potentially less optimal than model-based predictions, have the advantage of consistency and reliability. They can provide a straightforward, possibly costlier, solution in terms of operational efficiency but ensure that the control system remains functional in all scenarios. This approach has been described successfully in [10], where a rule-based algorithm is used as a fallback when the learning-based MPC does not yield a feasible solution.

These alternatives, while effective in ensuring system operation under all conditions, come with their own set of trade-offs:

- Direct MILP solution: higher computational costs and longer solution times, which could be prohibitive in real-time operational scenarios.

- Rule-based strategy: Potential loss in optimality but increased reliability and simplicity in implementation.

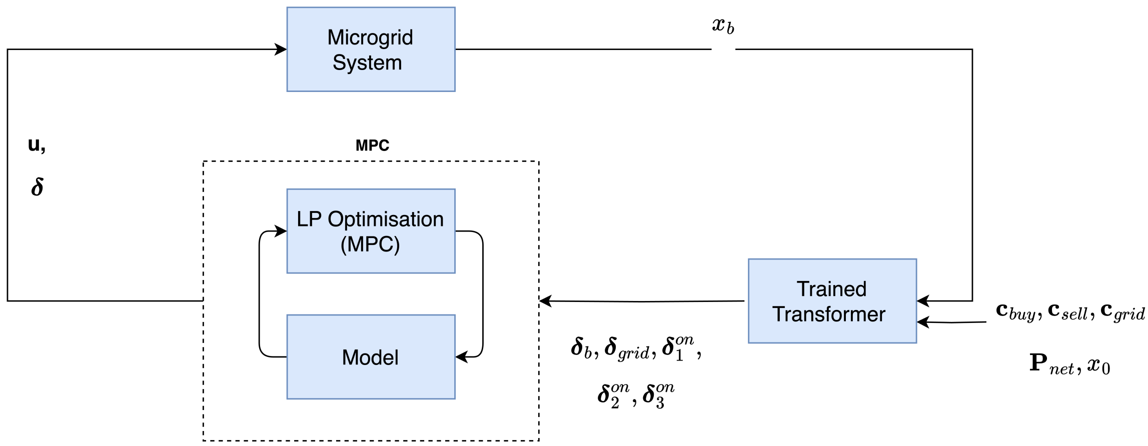

Figure 4-4: Transformer-based MPC diagram for microgrid

Deciding on the best approach depends on specific operational criteria, including the criticality of decision speed, acceptable levels of operational costs, and the typical complexity of the scenarios encountered. Further research and testing may be required to balance these factors optimally for the microgrid application.

In this section, it became apparent that despite high accuracy and a satisfactory success rate, the computation time required by the transformer poses a significant challenge. Addressing this requires exploring optimisation strategies to mitigate inference time and enhance system performance. In Appendix B, we explore a method to address this computation time and discuss other potential strategies to address the computational challenges encountered.

4-4 Closed Loop MPC Simulations

In this section, we detail the outcomes of closed-loop MPC simulations, designed to compare the traditional hybrid MPC approach with the transformer-based MPC technique. In the previous sections, the simulations were performed in open-loop under fixed conditions. In contrast, the closed-loop approach dynamically updates the microgrid's state based on system feedback, directly responding to the inputs applied during each simulation step. This approach adjusts the state with each time step and progresses through the microgrid data over time, thereby incorporating an additional future time step at each iteration. This setup offers a more authentic simulation environment, designed to mirror actual operational conditions closely, thereby allowing us to evaluate the transformer-based optimisation's performance in realistic scenarios. Figure 4-4 depicts the control scheme used in this section.

Simulation Transformer-Based MPC

We first analyse the performance of the transformer-based MPC over a shorter horizon. During the evaluation, we focused on average costs, average computational time, and the number of infeasible configurations during operations (the success rate). When the MPC predicts an

Table 4-6: Comparison of closed Loop MPC simulations

Metric	Optimal MPC	Transformer-based MPC
Avg. Costs	3786.50	3801.53 [0.3%]
Avg Computation Time [s]	31.38	87.33
Avg. Infeasible [%]	0%	19%

infeasible configuration, we fall back to the optimal MPC to ensure the simulation runs to completion as detailed in Section 3-5.

For the simulation of the transformer-based MPC, we use the same trained transformers described earlier in this chapter and the data we used earlier. We use the validation dataset of 3200 samples based on realistic microgrid data as described in Section 3-2-1 to ensure that the transformer-based MPC cannot predict directly based on training samples. The simulations are performed over 150 time steps, and the average results are taken over all simulations. In total, we perform 150 simulations with different initial states and price and load profiles to get a good insight into the performance of the transformer-based MPC. We use a prediction horizon of: $N_{\mathrm{P}} = 25$, corresponding to approximately 12h. Table 4-6 displays the results of the MPC simulations.

From Table 4-6, it can be seen that the transformer-based MPC works mostly as expected when we consider the earlier conducted experiments. The transformer-based MPC has a slightly higher cost on average, which should be expected as we compare it to an optimal controller using Gurobi. The optimality gap is lower than that found in open-loop, which is to be expected since the optimal controller has taken over as a fallback in part of the time steps, reducing the optimality gap.

The percentage of time that the transformer-based MPC is infeasible is higher than in the open loop case, which was not particularly expected. One explanation can be that the feedback MPC computes states that were underrepresented or not represented in the training data. This can make predictions more difficult for the transformer-based MPC and lead to infeasible solutions. During the training phase, the range of the initial battery state inputted into the transformer varies from 25 to 250 kWh. However, we observe that the battery state can extend outside these predefined ranges in simulation scenarios. During closed-loop simulations, we use the current system state as state input to the transformer. This may affect the predictive performance of the transformer-based MPC's predictions in the closed-loop MPC. One way to resolve and test this issue would be to make a more diverse set of (initial) states for the transformer during training, including more varied operational situations and retrain the transformer.

Further, a notable aspect observed during simulations is that when the transformer-based MPC fails to predict an optimal state for a specific timestep, it often generates infeasible solutions for subsequent timesteps. This may be because the data differs only slightly from time step to time step. This trend, where slight variations in state, price, and load data occur, can increase the number of consecutive infeasibilities and yield a higher infeasibility rate.

Lastly, we observe a discrepancy between the computation times of the closed-loop MPC and open-loop MPC, primarily due to differences in how we measured these times during simulations. For the open-loop MPC, computation time was explicitly recorded for the runtime and inference time directly associated with solving the MILP. In contrast, for the closed-loop MPC, we measured the computation time for the entire sequence of time steps, which includes additional processing beyond just solving the MILP. This methodological difference explains why computation times reported for the closed-loop MPC are not directly comparable to those from the open-loop MPC. Despite this, we note that the transformer-based MPC consistently shows longer computation times than the optimal MPC, aligning with our expectations based on previous results. This longer processing time reflects the inherent computational complexity introduced by the transformer's sequence processing, especially when adjusting to dynamic system states in real-time.

4-5 Comparison of Transformer-Based MPC Against Benchmark LSTM

Progressing from evaluating cascaded transformers, we assess the performance of the transformer-based MPC to another leading sequence modelling technique. Both transformers and long short-term memory (LSTM) have excelled in sequence modelling tasks over recent years, making them prime candidates for a comparative analysis in the context of MPC. The relevance of LSTMs in similar applications is shown in [18], where LSTMs are also used to predict binary sequences in MILPs for different applications in a similar manner.

One significant advantage in comparing transformers to LSTM approaches is that they can handle similar data preprocessing methods and formats. This uniformity allows for a focused comparison primarily based on the differences in their architectures rather than variations in input handling. By maintaining consistent data treatment, we ensure that any observed performance differences can be attributable to the intrinsic qualities of each approach rather than external factors. This approach allows for a clearer assessment of which architecture might better suit the complex needs of microgrid management under the MPC framework.

LSTMs, a specific variant of recurrent neural networks, were developed to address the challenge of vanishing and exploding gradients in sequence modelling with remarkable results. For an in-depth discussion of the LSTM architecture and its operational mechanisms, we refer to [30].

Experimental Setup

We have chosen to use an open-loop setting to evaluate the predictive accuracy of transformer-based MPC versus LSTM-based MPC. This approach allows us to directly compare the prediction capabilities of the different approaches without external influences from feedback mechanisms.

The LSTM and MPC are trained on the same amount of data using the same dataset, ensuring any performance differences are due to the approaches themselves and not variations in training data. Also, we can use the same data preprocessing and representation, i.e. making tensors of the relevant data, normalising the data and splitting it into mini-batches for efficient training. The hyperparameters of the LSTM are tuned based on empirical results and are tuned to achieve good performance to make the comparison as fair as possible.

Table 4-7: Performance comparison of LSTM and transformer

Metric	LSTM	Transformer	Gurobi
Training Time [s]	1796.84	891.88	-
Success Rate [%]	99.80	99.40	100
Optimisation Time [s]	0.0021	0.0487	0.0037
Optimality Gap [%]	0.18	0.76	-

To evaluate the performance of both, we use several measures. First, we assess the training time. This gives insight into the computation efficiency during training. After training, both models are tested with new sequences from a validation set to measure their performance and see how well both models generalise. The models are compared regarding prediction time, optimality gaps, and success rate during prediction.

Evaluation of LSTM vs Transformer in Open Loop

Table 4-7 provides a comparative overview of the key performance measures for both models. Note that we use the cascaded transformer setup and that, from experience, we do not expect that it is necessary that the LSTM requires a cascaded setup.

From the table, it can be seen that training an LSTM takes longer compared to a transformer. This could be expected as the transformer architecture, in general, is very efficient in training due to its parallel processing capabilities and self-attention mechanism, which requires simple computations. However, it should be noted that the entire transformer-based MPC takes longer to train, as we chose to train multiple transformers for the cascaded setup. We can train multiple transformers with different configurations in parallel, so we compare the training time of the single transformers in this evaluation.

When comparing the performance of the transformer-based MPC to the LSTM-based MPC, we observe that our transformer-based method cannot outperform the benchmark LSTM. Both perform adequately in the open-loop setting; the success rates are similar, and the optimality gaps are also low. Although the LSTM outperforms the transformer in both metrics. The main difference between the two is the average time required for optimisation. As we observed earlier, the current transformer is inefficient in prediction and, therefore, takes too long, which causes the transformer-based MPC to take significantly more time than the LSTM and Gurobi.

Conclusion of the Benchmark Comparison

The evaluation of LSTM and transformers for predicting sequences in microgrid applications reveals the distinct strengths and limitations of each approach. Although transformers do not match the LSTM in prediction efficiency, they achieve comparable success rates and optimality gaps. This indicates that while the fundamental architecture of transformers holds promise, the extended prediction time—primarily due to the current computational setup—presents a significant bottleneck. Unlike LSTMs, which are less parameter-dense and do not rely heavily on parallel computing, transformers are more susceptible to performance inefficiencies in environments that do not fully leverage their parallel processing capabilities, such as typical CPU-based systems, which were available for this experiment.

Future efforts should focus on optimising the computational aspects of transformer inference to enhance the practicality of using transformers in microgrid data prediction. Streamlining these processes could potentially elevate the transformer to a more competitive position relative to the LSTM, especially when rapid prediction capabilities are crucial.

Moreover, considering that LSTMs inherently manage fewer parameters and thus operate more effectively within CPU-constrained environments, it becomes essential to adapt transformers to be less resource-intensive or shift computational strategies to platforms capable of fully exploiting their architecture. The insights from this benchmark evaluation suggest that while LSTMs currently excel in efficiency and adaptability to less powerful computational environments, transformers—with targeted optimisations—could eventually harness their inherent capabilities for parallel processing to deliver enhanced performance.

4-6 Discussion

This section delves into the implications and significance of the findings presented in this chapter. By analysing the outcomes of various simulation scenarios, this discussion aims to interpret the performance of the transformer in microgrid control, assessing its practical viability and potential integration into existing systems. We evaluate the transformer's ability to meet the operational demands of microgrids, considering the computational challenges and the comparative effectiveness against other advanced machine learning techniques and traditional methods.

Transformer prediction time
The transformer has demonstrated its capability in accurately predicting binary decisions, which is crucial for the optimisation and efficiency of MPC in microgrids. However, its computational efficiency, particularly in prediction times, faces challenges when compared directly with commercial solvers like Gurobi or benchmarks such as LSTMs. The implementation in ONNX runtime in Appendix B has notably improved the transformer's inference time, enhancing its potential for real-time applications and demonstrating that strategic implementation changes can significantly impact prediction speed. These improvements suggest that the model's efficiency could substantially increase under optimised conditions.

Moreover, the comparison between the transformer and Gurobi requires careful consideration. Gurobi shows superior computational efficiency, particularly due to the optimised resources used in the study. We are evaluating a transformer developed in Python with PyTorch, which contrasts with Gurobi —a highly optimised commercial solver ideally suited for CPU applications. The transformer's computations would greatly benefit from parallel processing in GPU units or more optimised computational environments than Python. This discrepancy in the underlying technology can significantly influence performance, especially in computational efficiency. Therefore, while insightful, these comparisons must be critically evaluated to ensure a fair and meaningful assessment of the transformer's potential in real-world applications. With further consideration of the implementation and environment, it is expected that we can improve the prediction time of the transformer and create a fairer comparison.

Longer Prediction Horizons
Challenges observed with longer horizon predictions underscore model enhancements' need to handle extended sequences effectively. The transformer demonstrated high accuracy and reliability over shorter horizons but faced significant performance drops for longer horizons. This issue highlights a potential oversight in our initial approach — we assumed that the same amount of data would suffice for both short and twice-as-long prediction horizons. However, longer horizons likely involve more complexities and dynamics, suggesting that a larger or more diverse dataset might be necessary to improve model training and performance.

The lower success rate of the low-complexity transformer configuration in longer horizon predictions highlights the need for model tuning. Increasing the model's complexity led to significant performance improvements, demonstrating its potential to handle complex, extended sequences better. However, this benefit must be balanced against increased computational demands, as greater complexity can prolong prediction times, limiting the model's real-time application. Therefore, future refinements should include strategic adjustments to the model architecture and the training data's volume and diversity. This ensures the transformer remains effective and efficient for real-world use, where accuracy and operational speed are crucial.

Cascaded Transformer Strategy
Adopting a cascaded model approach has significantly addressed the reliability issues previously noted in success rates. By employing multiple trained transformers in sequence, we've seen a substantial improvement in the success rate, enhancing the online applicability of the transformer in MPC. This strategy could also be tailored to train transformers for specific operational scenarios within the microgrid, e.g. periods of low renewable energy generation, further increasing the reliability. However, it is important to note that while this approach enhances success rates, it can also increase computational demands. Specifically, if the MPC frequently needs to fall back on backup transformers, this can result in higher overall prediction times, potentially impacting the system's efficiency.

Closed-Loop MPC Performance
Evaluating the transformer within a closed-loop MPC setting highlighted its adaptability to dynamic operational conditions. This testing phase crucially assessed how system feedback influences subsequent predictions and how the model manages continuous state updates. Challenges such as the occurrence of infeasible solutions and their propagation through multiple time steps were observed, which critically impacts the system's reliability. Fallback strategies, like switching to an optimal MPC or a rule-based system, played a significant role in maintaining operational continuity. Despite these strategies, the transformer displayed a higher infeasibility rate than open-loop scenarios, suggesting the need for broader training datasets that include more variable operational states, making the transformer more adept in dynamic changes in the microgrid. Lastly, the optimality gap of the closed-loop MPC is considered slightly too low since the fallback controller produced optimal solutions when fallback was needed.

By adopting these strategies, future research can aim to overcome the current limitations and fully exploit the potential of transformers in energy management systems. Specific recommendations are given for further research in Chapter 5.

4-7 Conclusion

Reflecting on the simulations and analyses conducted, this chapter concludes by exploring the potential of utilising transformers for microgrid control within an MPC framework. The conclusions drawn from this chapter highlight both the potential and the challenges of implementing transformers in this specific application. The predictive capabilities of transformers have shown high accuracy and reliability, particularly for shorter prediction horizons, promising significant enhancements in microgrid operations. However, the extended prediction times in their current configuration present substantial challenges for their practical application in real-time MPC settings.

These findings indicate that while transformers can deliver precise predictions for discrete control actions, transformers exhibit extended prediction times in their current configuration that challenge their practicality for real-time applications in MPC. The time required to predict binary sequences using the transformers is too lengthy, making their direct implementation in online MPC for MILP problematic. Despite these initial setbacks, substantial improvements appear feasible through alternative implementations. In Appendix B, we showed that implementing the transformer to an ONNX format could streamline the computation process, significantly reducing prediction times. Additionally, deploying the transformer on GPU-accelerated hardware or exploring other advanced computational environments is expected to enhance performance, making the use of transformers more viable for these applications. Such strategies hold promise for achieving the full potential of transformers in online MPC scenarios.

From the longer horizon MPC simulations we conclude that further research is needed to assess the transformer's effectiveness with longer sequences. Transformers are known for their ability to handle longer, more complex sequences well. However, our experiments applying the transformer in longer horizon MPC indicate a higher rate of infeasible solutions and long prediction times. We observed that further hyperparameter tuning is crucial to improve performance for the longer-horizon MPC and conclude that revising the diversity and amount of data can be beneficial. Addressing this is particularly relevant because computational demands typically increase with longer horizons, making learning-based methods specifically needed for these scenarios.

With further development and implementations in different computation environments, transformers can potentially enhance predictive control strategies in microgrids, making them more efficient. Continued research into computational optimisations and tuning is essential to make transformers applicable for online MPC.

Chapter 5

Conclusion and Future Work

This chapter covers the main contributions of this thesis and the conclusions that can be drawn from the thesis. We conclude with suggestions for future work.

5-1 Contributions

The research question central to this thesis is formulated as:

How can transformers be applied to predict the optimal binary solution of Mixed-Integer Linear Programming problems, specifically within the context of Mixed Logical Dynamical systems and learning-based Model Predictive Control?

To address this question, we adapted the transformer architecture—originally designed for natural language processing—to predict binary solutions of MILP problems within MPC. This adaptation faced considerable challenges due to microgrid data's continuous nature, which significantly differs from the discrete and static data typically processed in natural language tasks.

The first contribution of this research is the development of data preprocessing techniques that transform real-time, variable energy data streams into a format suitable for transformer processing. This required applying feature embedding techniques, which were inspired by advancements in fields such as vision transformers and different machine learning applications. It enabled us to effectively handle the unique characteristics of energy data within the microgrid system with the transformer.

Another contribution is the application of the transformer to simplify complex mixed-integer linear programming (MILP) problems into more tractable linear programming (LP) problems. By accurately predicting binary decision sequences, the transformer substantially reduced the dimensionality and complexity of the decision space, which leads to an increase in computational efficiency and scalability of model predictive control (MPC) operations. This simplification enables more responsive and cost-effective energy management in microgrids. We further showed how the transformer predictor can seamlessly be integrated into the MPC framework.

We also introduced the cascaded transformer strategy, an approach that significantly improved the reliability and success rate of the predictions. By employing multiple trained transformers in sequence, this strategy improved the reliability of the transformer-based approach. It demonstrated its potential to be customised for specific operational scenarios within the microgrid, enhancing the online applicability of transformer-based MPC.

We set up various tests and validated our approach through a series of simulations against traditional MPC methods and benchmark long short-term memory (LSTM)-based MPC. These validations demonstrated the accuracy and efficiency of the transformer-based MPC approach. They provided a benchmark for assessing the relative advancements contributed by the transformers in managing the dynamic environments typical of microgrid systems. It also highlighted the potential and challenges of transformers in the context of learning-based MPC.

5-2 Conclusion

Reflecting on the studies and simulations conducted, this chapter synthesises the insights gained and outlines the implications of employing transformers for microgrid control within the MPC framework. The research explored in this thesis demonstrates the transformer's capacity for enhancing microgrid operations. It also identifies significant challenges that must be addressed to achieve its full potential in practical applications.

Adapting the transformer architecture to microgrid management represents an advancement in applying advanced machine-learning techniques to control energy systems, specifically microgrids. By effectively processing continuous, dynamically changing microgrid data and converting complex MILP problems into more tractable LP problems, the transformer has shown it can streamline the decision-making process within MPC. This capability enhances the computational efficiency of MPC systems, potentially facilitating faster and more cost-effective energy management solutions.

However, while the transformer has exhibited high accuracies of below 1% and reliability of up to 99.4%, particularly for shorter prediction horizons, its application in online MPC settings presents substantial challenges. The primary concern is the extended prediction times observed during the experiments, which limit the transformer's applicability for online MPC. Through an example, we suggested potential pathways to mitigate these issues, including adopting ONNX runtime to improve computational efficiency. Other ways could be exploring GPU-accelerated hardware or other advanced computational environments since we used CPU and Python during the research, which is not optimal for the application.

Moreover, the research highlighted the need for further studies to optimise the transformer's performance with longer prediction horizons and closed-loop MPC. The issues of increased infeasible solutions and lengthy prediction times for extended sequences suggest that additional model tuning and a more diverse training dataset are needed to enhance the transformer's effectiveness. Addressing these challenges is crucial as computational demands typically escalate with longer horizons, precisely the scenarios where learning-based methods like transformers could provide significant benefits.

Through developing and refining transformers for microgrid control, transformers can be promising for predictive control strategies. Further research into computational optimisations, model tuning, and integrating transformers in different computational environments is essential. These efforts will ensure that transformers can meet the dynamic demands of modern energy systems, providing a tool for microgrid MPC.

5-3 Recommendations for Future Work

As this research concludes, it is evident that while transformers hold promise in enhancing MPC in microgrids, several challenges and opportunities for advancement remain. The following suggestions for future work aim to build upon the achievements of this thesis, addressing the identified limitations and exploring transformers further within MPC for microgrids. These recommendations are to refine the capabilities of transformers further, enhance their operational efficiency, and extend their applicability in real-time energy systems management:

1. Enhancing real-time capabilities: Investigate advanced computational techniques to reduce the prediction time of transformers without compromising accuracy. Explore hardware acceleration options like GPU utilisation, and consider implementing the transformer in more suitable environments than Python.

2. Extending model complexity and Depth: experiment with increasing the complexity of transformer models to improve their capability for longer horizon predictions. Assess the trade-offs between model complexity and computational efficiency to find an optimal balance that maximises prediction accuracy and operational speed.

3. Cascaded transformer exploration: further refine the cascaded transformer strategy to improve efficiency and reduce the need for fallbacks. Develop adaptive mechanisms that dynamically select the most appropriate transformer model based on the current operational context (e.g. weather and available resources). This would not only increase reliability but is expected to reduce the optimality gap further as well.

4. Data diversity and training enhancements: expand the training datasets to include a wider range of operational scenarios. Implement advanced data augmentation techniques to enhance the diversity of training data artificially, improving the models' generalisability. This is expected to be explicitly beneficial for the longer horizon MPC applications.

These areas for future research are aligned with the work presented in the thesis, aiming to enhance the integration of advanced transformers into microgrid control.

Appendix A

Binary Decision Conversion Data Preprocessing

Table A-1: Conversion of Integer Values to 5-bit Binary Representations

Integers 0 to 15

Integer	Binary Representation
0	00000
1	00001
2	00010
3	00011
4	00100
5	00101
6	00110
7	00111
8	01000
9	01001
10	01010
11	01011
12	01100
13	01101
14	01110
15	01111

Integers 16 to 31

Integer	Binary Representation
16	10000
17	10001
18	10010
19	10011
20	10100
21	10101
22	10110
23	10111
24	11000
25	11001
26	11010
27	11011
28	11100
29	11101
30	11110
31	11111

Appendix B

Computational Optimisation: ONNX Runttime

In Section 4-3, it became apparent that despite high accuracy and a satisfactory success rate, the computational time required by the transformer poses a significant challenge. Addressing this requires exploring optimisation strategies to mitigate inference time and enhance overall system performance.

Open Neural Network Exchange (ONNX) Runtime, an open-source scoring engine for ONNX models, offers a potential solution. Supporting a broad spectrum of optimisation techniques, including hardware acceleration and pre-trained optimisations, ONNX Runtime is particularly effective in enhancing transformer inference across diverse platforms, thus significantly reducing inference times [39]. Research works further support this, e.g. [40].

Implementing ONNX Runtime for Transformer-Based model predictive control (MPC)

To address the computational efficiency of the transformer for microgrid control, we implemented ONNX Runtime to see the effects on inference time. The first step involved exporting the trained transformer to the ONNX format. This conversion process maintains data preprocessing consistency, ensuring consistency in predictions. We then assessed the performance of both single and cascaded transformers implemented in ONNX against their PyTorch counterparts and traditional mixed-integer linear programming (MILP) approaches, focusing on success rates, optimality gaps, and computation times.

Performance Evaluation

The evaluation aimed to highlight the computational benefits brought by the implementation of ONNX Runtime. Significant improvements were noted, especially in computation time, which is crucial for real-time operations.

Table B-1: Comparison of ONNX Runtime, single transformer, and Gurobi

Metric	Regular Transformer	Transformer in ONNX	Gurobi
Success Rate [%]	88.30	87.10	100
Optimality Gap [%]	0.36	1.28	-
Avg Computation Time [s]	0.0571	0.0025	0.0041

Table B-2: Comparison of ONNX Runtime, cascaded transformer, and Gurobi using fallback

Metric	Regular Transformer	Transformer in ONNX	Gurobi
Success Rate [%]	97.10	96.60	100
Optimality Gap [%]	0.91	1.76	-
Avg Computation Time [s]	0.0508	0.0024	0.0041

Although the transformers implemented in ONNX show a slightly lower success rate and a higher optimality gap than their regular counterparts, the reduction in computation times is remarkable, particularly when employing cascaded transformers. This underscores the efficiency gains achieved through ONNX Runtime, supporting the feasibility of using transformers without substantial delays.

The tables show slight differences in success rates and optimality gaps compared to the regular transformers. This was not as expected since the adoption of ONNX Runtime primarily involves differences in implementations of the same transformers. These discrepancies could be due to ONNX Runtime's optimisation processes. While designed to enhance computational efficiency, modifications such as graph optimisation and layer fusion can subtly alter the transformer's behaviour. Fixed execution paths and changes in numerical precision intended to boost processing speeds might affect the transformer's output, leading to variations in performance metrics. Furthermore, ONNX runtime is an open-source API that is continuously being developed. Therefore, although unlikely, faults or inaccuracies may need further development, causing the performance to differ slightly.

Implications and Alternative Strategies

This ONNX case study is just one example of computational optimisation and how implementation can greatly affect the prediction time of the transformer strategy. Other strategies, such as code optimisation or more efficient programming languages like C#, could further reduce computational demands. Advanced compilation techniques or frameworks might also offer additional benefits.

This demonstration proves that substantial computational improvements are achievable, which is crucial for integrating transformer-based MPC into real-time applications. Future research might explore combining these strategies to find the best balance between accuracy and efficiency, tailoring solutions to the specific operational needs within microgrid management systems.

Appendix C

System Equations

This appendix describes the MLD formulation used for the microgrid in the thesis [10]:

C-1 Energy Storage System (ESS)

$$x_b(k+1) = \begin{cases} x_b(k) + \frac{T_s}{\eta_d} P_b(k) & \text{if } P_b(k) < 0 \\ x_b(k) + T_s \eta_c P_b(k) & \text{if } P_b(k) \geq 0 \end{cases} \tag{C-1}$$

- $x_b(k)$: energy stored in ESS
- η_c,η_d: charging and discharging efficiencies
- $P_b(k)$: Power exchanged with ESS
- T_s: Sampling interval

With the introduction of an auxiliary variable, we obtain:

$$x_b(k+1) = x_b(k) + T_s \left(\eta_c - \frac{1}{\eta_d} \right) z_b(k) + \frac{T_s}{\eta_d} P_b(k), \tag{C-2}$$

- $z_b = \delta_b(k) P_b(k)$
- $\delta_b(k) = 1 \leftrightarrow P_b(k) \geq 0$

C-2 Variables

$$\delta = \begin{bmatrix} \delta_b \\ \delta_{grid} \\ \delta_1^{on} \\ \delta_2^{on} \\ \delta_3^{on} \end{bmatrix}, \quad z = \begin{bmatrix} z_b \\ z_{grid} \end{bmatrix} \quad u = \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix}$$

Table C-1: Microgrid Parameters

PARAMETER	VALUE
Maximum ultracapacitor energy level x_{uc}	50 [kWh]
Minimum ultracapacitor energy level x_{uc}	2 [kWh]
Maximum battery energy level x_b	250 [kWh]
Minimum battery energy level x_b	25 [kWh]
Battery charging efficiency $\eta_{c,b}$	0.90
Battery discharging efficiency $\eta_{d,b}$	0.90
Ultracapacitor charging efficiency $\eta_{c,uc}$	0.99
Ultracapacitor discharging efficiency $\eta_{d,uc}$	0.99
Maximum interconnection power flow limit P_{grid}	1000 [kW]
Minimum interconnection power flow limit P_{grid}	-1000 [kW]
Number of generators N_{gen}	3
Maximum power providable by the battery P_b	100 [kW]
Maximum power injectable to the battery P_b	-100 [kW]
Maximum power providable by the ultracapacitor P_{uc}	25 [kW]
Maximum power injectable to the ultracapacitor P_{uc}	-25 [kW]
Maximum power level of the dispatchable generators P_{dis}	150 [kW]
Minimum power level of the dispatchable generators P_{dis}	6 [kW]

C-3 MLD Constraints

MLD form constraints:

$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \tag{C-3}$$

, where $\delta \in \mathbb{Z}^5$, $z \in \mathbb{R}^2$, $u \in \mathbb{R}^5$ $x \in \mathbb{R}^1$

- **State, initial condition**

$$\underbrace{\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}}_{E_4} x_b(k) \leq \underbrace{\begin{bmatrix} x_0 \\ x_0 \\ 250 \\ -25 \end{bmatrix}}_{E_5}$$

- **Input**

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{E_1} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix} \leq \underbrace{\begin{bmatrix} 100 \\ 100 \\ 1000 \\ 1000 \\ 150 \\ -6 \\ 150 \\ -6 \\ 150 \\ -6 \end{bmatrix}}_{E_5}$$

- **Continuous Auxiliary Variables**

$$\underbrace{\begin{bmatrix} -M_b & 0 & 0 & 0 & 0 & 0 \\ m_b & 0 & 0 & 0 & 0 & 0 \\ -m_b & 0 & 0 & 0 & 0 & 0 \\ M_b & 0 & 0 & 0 & 0 & 0 \\ 0 & -M_g & 0 & 0 & 0 & 0 \\ 0 & m_g & 0 & 0 & 0 & 0 \\ 0 & -m_g & 0 & 0 & 0 & 0 \\ 0 & M_g & 0 & 0 & 0 & 0 \end{bmatrix}}_{E_2} \begin{bmatrix} \delta_b \\ \delta_{grid} \\ \delta_1^{on} \\ \delta_2^{on} \\ \delta_3^{on} \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}}_{E_3} \begin{bmatrix} z_b \\ z_{grid} \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{E_1} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ -m \\ M \\ 0 \\ 0 \\ -m \\ M \end{bmatrix}}_{E_5}$$

, where $m_b = -100\ kW$, $M_b = 100\ kW$, $m_g = -1000\ kW$, $M_g = 1000\ kW$.

- **Discrete Variables**

ESS and Grid:

$$\underbrace{\begin{bmatrix} -m_b & 0 & 0 & 0 & 0 & 0 \\ -(M_b + \epsilon) & 0 & 0 & 0 & 0 & 0 \\ 0 & -m_g & 0 & 0 & 0 & 0 \\ 0 & -(M_g + \epsilon) & 0 & 0 & 0 & 0 \end{bmatrix}}_{E_2} \begin{bmatrix} \delta_b \\ \delta_{grid} \\ \delta_1^{on} \\ \delta_2^{on} \\ \delta_3^{on} \end{bmatrix} \leq \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{E_1} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix} + \underbrace{\begin{bmatrix} -m_b \\ -\epsilon \\ -m_g \\ -\epsilon \end{bmatrix}}_{E_5}$$

, where $m_b = -100\ kW$, $M_b = 100\ kW$, $m_g = -1000\ kW$, $M_g = 1000\ kW$.

Generators:

$$\underbrace{\begin{bmatrix} 0 & 0 & -m_{gen} & 0 & 0 \\ 0 & 0 & -(M_{gen}+\epsilon) & 0 & 0 \\ 0 & 0 & 0 & -m_{gen} & 0 \\ 0 & 0 & 0 & -(M_{gen}+\epsilon) & 0 \\ 0 & 0 & 0 & 0 & -m_{gen} \\ 0 & 0 & 0 & 0 & -(M_{gen}+\epsilon) \end{bmatrix}}_{E_2} \begin{bmatrix} \delta_b \\ \delta_{grid} \\ \delta_1^{on} \\ \delta_2^{on} \\ \delta_3^{on} \end{bmatrix} + \leq \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{E_1} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix} + \begin{bmatrix} -m_{gen} \\ -\epsilon \\ -m_{gen} \\ -\epsilon \\ -m_{gen} \\ -\epsilon \end{bmatrix}$$

, where $m_{gen} = 6\ kW$ and $M_{gen} = 150\ kW$

- **Power Balance**

$$\underbrace{\begin{bmatrix} 1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{E_1} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix} + \underbrace{\begin{bmatrix} P_{load} - P_{res} \\ P_{res} - P_{load} \end{bmatrix}}_{E_5} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- **Generator constraint**

$$\underbrace{\begin{bmatrix} 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -150 & 0 & 0 \\ 0 & 0 & 0 & -150 & 0 \\ 0 & 0 & 0 & 0 & -150 \end{bmatrix}}_{E_2} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{E_1} \begin{bmatrix} P_b \\ P_{grid} \\ P_1^{dis} \\ P_2^{dis} \\ P_3^{dis} \end{bmatrix}$$

Appendix D

Paper

This appendix includes the paper entitled "Learning-Based Control of Microgrids with Transformers and Model Predictive Control" which details the experimental procedures and results from the thesis. The paper is included in its entirety on the following pages.

Learning-Based Control of Microgrids with Transformers and Model Predictive Control

Bob van der Woude MSc Student Systems and Control, Delft University of Technology

***Abstract*—Microgrids are pivotal for enhancing energy efficiency and sustainability within modern energy systems. By integrating renewable energy sources and operating independently or alongside the main grid, microgrids reduce reliance on centralised power supplies and improve resilience to disruptions. This research addresses the unit commitment problem in microgrids, modelled as mixed logical dynamical systems, through Model Predictive Control (MPC). The challenge of MPC lies in its requirement to solve computationally intensive mixed-integer linear programming (MILP) problems. To address this, we introduce transformers that learn the binary decisions of MILP problems offline, allowing for the simplification of these problems to more manageable linear programming problems during online MPC. This integration enhances MPC's computational efficiency and scalability, with simulations demonstrating efficacy in predicting the binary components of the MILP problems to find near-optimal control decisions.**

I. Introduction

A. Introduction

The pursuit of sustainable energy solutions remains a pivotal challenge as the global demand for energy rises alongside growing environmental concerns. Microgrids, as innovative energy solutions, have shown promise by offering both a mitigation of environmental impact and an enhancement of energy resilience. These decentralised energy networks stand out for their unique capability to integrate seamlessly with renewable sources, such as solar and wind, and their ability to operate connected and reconnected from the centralised electrical grids. Key international reports, such as those from the International Renewable Energy Agency (IRENA), underscore the pivotal role of renewable energy, suggesting it has the potential to satisfy up to two-thirds of the world's energy needs. This significant potential places microgrids at the heart of strategies aimed at reducing global greenhouse gas emissions to meet temperature increase targets set by the Paris Agreement on climate change [1], [2],[3].

Microgrids contribute to environmental goals and ensure the stability and security of energy supply, especially in regions prone to extreme weather conditions or geographical isolation from central power systems. By facilitating a more stable and reliable energy distribution, microgrids help mitigate the vulnerabilities associated with energy production and distribution, thus reinforcing the overarching framework of global energy security [4].

Bob van der Woude is a master's student at TU Delft, The Netherlands.

B. Problem Formulation

Within the operational dynamics of microgrids, the unit commitment problem emerges as a significant challenge. This problem involves a complex mathematical optimisation that necessitates the coordination of diverse energy production units to meet demand while minimising operational costs efficiently [5]. Model Predictive Control (MPC) stands out as a strategy due to its predictive capabilities and robust mechanism for managing variable constraints, making it suitable for handling the complexities inherent in microgrid operations [6]. Despite its advantages, the real-time application of MPC in systems such as microgrids is hindered by the need to solve computationally intensive Mixed-Integer Linear Programming (MILP) problems at each control iteration due to the presence of continuous and discrete variables. These $\mathcal{NP}$-hard problems introduce considerable computational burdens, posing formidable challenges for real-time microgrid management [7].

This study proposes an innovative learning-based approach to the MPC framework to navigate these challenges. In several works, learning-based methods have shown successful results in MPC, e.g. [8], [9]. Since MPC relies on sequential decision-making to predict and adjust the system's decisions over time, transformers can be particularly well-suited for this context. Renowned for their efficiency in processing sequential data through advanced self-attention mechanisms, transformers provide a promising avenue for simplifying the combinatorial MILP decision-making processes [10]. By effectively learning these binary decisions, transformers can reduce complex MILP problems into more tractable linear programming (LP) problems, significantly reducing computational intensity. This adaptation enhances the feasibility of deploying MPC in real-time control of microgrids. It leverages the sequential nature of data within the MPC framework to improve decision accuracy and system responsiveness.

The paper begins with a detailed description of microgrid systems in Section II, focusing on their components and dynamics. Section III explores using MPC for microgrid management. The role of transformers in improving MPC's efficiency is examined in Section IV. Sections V and V-F cover methodological details for implementing the transformer. Simulation results comparing traditional and transformer-based MPC are analysed in Section VI. The paper concludes in Section VII with key findings and future research directions, emphasising the impact of machine learning on energy management. The programming codes used in this study are provided in [11].

II. System Description

This section outlines the microgrid system's dynamics and operational parameters based on the formulation established in previous research [8]. This setup includes energy storage, local generation units, a bidirectional connection to the main grid, and uncontrollable loads, which are essential for addressing the complexities of power dispatch in a microgrid environment.

A. Dynamic Modeling of Energy Storage Systems

The energy storage within the microgrid is modelled primarily as a battery system. It follows a hybrid dynamical law that monitors energy levels based on power exchange, whether charging or discharging. The state of charge at each step k, denoted by $x_b(k)$, is adjusted for efficiency losses during charging (η_c) and discharging (η_c) and the sampling interval, T_s [8]:

$$x_b(k+1) = \begin{cases} x_b(k) + \frac{T_s}{\eta_d} P_b(k) & \text{if} P_b(k) < 0 \\ x_b(k) + T_s \eta_c P_b(k) & \text{if} P_b(k) \geq 0 \end{cases} \quad (1)$$

here z_b is an auxiliary variable that modulates the power effect based on the storage system's mode of operation, indicated by the binary variable $\delta_b(k)$. This integration of binary and continuous variables is crucial for capturing energy storage dynamics and is described within the framework of Mixed-Logical Dynamical (MLD) systems [12].

B. Generator Units

The microgrid includes both renewable (zero-cost, uncontrollable) and dispatchable generators. Dispatchable generators provide adjustable power outputs, essential for responsive microgrid control, represented by the vector:

$$\mathbf{P}_{dis} = \left[P_1^{\text{dis}}(k) \dots P_{N_{\text{gen}}}^{\text{dis}}(k) \right]^T$$

C. Prices of Energy

The economic model of the microgrid incorporates distinct costs for energy: $c_{buy}(k)$ for purchasing from the main grid, $c_{sale}(k)$ for selling to it, and $c_{prod}(k)$ for local generation by dispatchable units. These prices are forecasted over a known horizon to assist in strategic planning.

D. Grid Interaction

The microgrid's interaction with the central grid is controlled through P_{grid} and the binary decision δ_{grid}, which dictate whether the microgrid imports or exports energy. The economic implications of these interactions are calculated by $C_{grid}(k)$:

$$C_{grid}(k) = \begin{cases} c_{buy}(k) \cdot P_{grid}(k) & \text{if} \delta_{grid}(k) = 1 (\text{importing}) \\ c_{sell}(k) \cdot P_{grid}(k) & \text{if} \delta_{grid}(k) = 0 (\text{exporting}) \end{cases} \quad (2)$$

III. Control of Microgrids using MPC

MPC employs a model-based strategy that utilises predictions about future system behaviour and input variables to optimise control actions over a defined horizon. It is well suited for the hybrid nature of microgrids, which feature continuous dynamics and discrete events since MPC has been extensively documented for its effectiveness in managing complex operational dynamics [12]. The adaptation of hybrid MPC formulations for microgrid control capitalises on MPC's ability to balance operational constraints with cost optimisation, providing a robust framework for the unit commitment problem which aims to manage the interplay of power generation, storage, and consumption in microgrids at minimal costs [13], [14],[15]. This highlights the strategic benefits of MPC in microgrid control and specifically addresses the challenges of the unit commitment problem.

A. Mathematical Formulation of the Control Problem

The objective of the MPC framework is to minimise the economic costs associated with energy production and exchange while satisfying load demands and operational constraints. This is formulated through the following objective function, which aims to minimise the combined costs of local energy production and grid interaction over a predictive horizon N_P:

$$\min_{\boldsymbol{P}_{dis}(k), P_{grid}(k), P_b(k), \boldsymbol{\delta}(k), \boldsymbol{z}(k)} J(\boldsymbol{P}_{dis}(k), C_{grid}(k), c_{prod}(k)) \quad (3)$$

The cost function J is defined as:

$$J(\boldsymbol{P}_{\text{dis}}(k), C_{\text{grid}}(k), c_{\text{prod}}(k)) = \sum_{j=0}^{N_p-1} \Bigg(C_{\text{grid}}(k+j) + c_{\text{prod}}(k+j) \sum_{i=1}^{N_{\text{gen}}} P_i^{\text{dis}}(k+j) \Bigg) \quad (4)$$

where N_p is the prediction horizon. The objective function is subjected to the constraints of the microgrid. The system dynamics of the microgrid model can be represented in the MLD formulation as mixed integer constraints.

B. System Constraints and MLD Formulation

The microgrid dynamics are encapsulated within an MLD system that converts the hybrid system behaviour into an MILP problem, serving as constraints within the MPC framework. These constraints are pivotal for ensuring that the microgrid operates within safe and efficient parameters:

$$E_1\boldsymbol{\delta}(k) + E_2\boldsymbol{z}(k) \leq E_3\boldsymbol{u}(k) + E_4 \tag{5a}$$

$$P_b(k) = \sum_{i=1}^{N_{gen}} P_i^{dis}(k) + P_{res}(k) + P_{grid}(k) - P_{load(k)} \tag{5b}$$

$$\underline{P}_b \leq P_b \leq \bar{P}_b \tag{5c}$$

$$\underline{P}_{grid} \leq P_{grid} \leq \bar{P}_{grid} \tag{5d}$$

$$\delta_i^{on}\underline{P}_{grid} \leq P_{grid} \leq \delta_i^{on}\bar{P}_{grid} \tag{5e}$$

$$\underline{x}_b \leq x_b \leq \bar{x}_b \tag{5f}$$

$$\boldsymbol{u}(k) = [\boldsymbol{P}_{dis}^{\top}(k), C_{grid}(k), P_b(k)]^{\top}$$

$$\text{for } i = 1, \ldots, N_{\text{gen}}$$

$$\text{for } k = 0, \ldots, N_p - 1$$

Constraint (5a) derives from the MLD formulation of the microgrid, ensuring the integration of logic and dynamics. Constraint (5b) maintains power balance, requiring all generated power to be either used within the system, stored, or exported. Constraints (5c) through (5f) specify operational limits: (5c) limits the power interactions with the battery; (5d) governs power exchanges with the main grid; (5e) regulates output from dispatchable generators; and (5f) sets the bounds on the battery's state of charge. These constraints collectively ensure the microgrid operates within its designed physical and operational parameters.

The challenges of implementing MPC within MLD systems include the $\mathcal{NP}$-hard nature of the MILP problems required to be solved at each control interval [16]. Addressing these computational challenges through advanced algorithms and machine learning techniques remains an active area of research, as MPC's potential for enhancing microgrids' operational efficiency and economic performance continues to be explored in depth, e.g. [8].

While the MPC framework provides a systematic approach to microgrid control, the computational demands of solving MILP problems at each control iteration pose significant challenges for real-time applications. To address these complexities, the integration of transformers is proposed. By leveraging their capability to process and predict sequences efficiently, transformers can simplify the computational tasks involved in MILP, potentially enhancing the real-time operational capabilities of MPC. This approach promises to streamline the computational process and retain the precision required for effective microgrid management. The subsequent section delves into how transformers can be applied to improve the scalability and efficiency of MPC in microgrid environments.

IV. Transformers

Transformers have revolutionised sequence modelling, particularly within fields like natural language processing (NLP), by addressing the limitations of earlier architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Vaswani et al. introduced the transformer architecture in "Attention is all you need" [10]. Transformers eliminate the need for sequential data processing, relying instead on self-attention mechanisms that allow parallel computation and have shown more effective and accurate handling of long-range dependencies.

A. Key Components of the Transformer Architecture

B. Key Features of the Transformer Architecture

The key components of the transformer architecture are described below. We refer to [10] for further elaboration on their exact functioning.

Self-Attention Mechanism: This mechanism enables the transformer to weigh the importance of different parts of the input sequence independently, enhancing the transformer's ability to process sequences in parallel. It calculates attention as a function of queries, keys, and values, significantly improving the efficiency and scalability of sequence modelling.

Positional Encoding: Unlike RNNs and LSTMs, transformers do not inherently process temporal data in sequence. Positional encodings are used to inject information about the order of the sequence into the transformer, allowing it to consider the position of data points within the sequence.

Multi-Head Attention: This extends the transformer's capability to focus on different parts of the sequence simultaneously, enabling it to capture various aspects of the sequence context more comprehensively.

C. Implications for Transformer-Based MPC in Microgrids

In MPC in microgrid systems, transformers potentially offer significant computational benefits due to their parallel processing capabilities, enhancing offline training and online control. During offline training, transformers rapidly handle large datasets, speeding up the training process. In online applications, transformers reduce decision-making times by predicting binary decisions, which is crucial for responding to fluctuating conditions in microgrids.

The self-attention mechanism in transformers enables precise prediction of binary decision sequences in MPC based on inputs like load demands, generation capacities, and costs. This improves the accuracy of operational decisions. Their scalability allows them to handle larger datasets and more complex networks without performance loss, supporting system expansion.

Further, the primary goal of our learning-based MPC approach is to forecast binary sequences that emerge from decision-making processes, influenced by past and anticipated future decisions and data. Recognising and harnessing this temporal structure is critical. The sequential nature of MPC typically reveals patterns in binary inputs that provide essential insights, making transformers, known for their excellence in sequence modelling, particularly well-suited for this task.

Integrating transformers into MPC frameworks leverages these advantages, transforming complex MILP problems into more manageable formats. This integration enhances the efficiency and responsiveness of energy management systems, facilitating quicker and more effective decision-making in real-time operations.

V. Methodology: Transformer for Binary Sequence Prediction

This section describes the methodology for developing and implementing a transformer to predict binary decisions in MILP formulations for microgrids. The goal is to enhance the computational efficiency of MPC by predicting the binary decision-making processes within MILP, utilising the transformer.

A. Data Collection and Preparation

Effective microgrid operation using MPC depends significantly on the accuracy of the input data, which reflects current and forecasted conditions like prices, load demands, and system states. We collect real-world and synthetically generated data to model these aspects. Synthetic data for energy costs are generated using normal distributions, reflecting realistic market variability. In contrast, actual load and generation data are sourced from Dutch main grid operations, scaled to fit microgrid capacities [17].

The following steps describe the method to collect data for the transformer:

1) Formulate N_{sim} exemplary MILP problems as described in Section III.
2) For each problem, extract the relevant parameters:

$$Z_N = \left\{ \left(x_0^{(i)}, \left\{c_{\text{buy}}^{(i)}(k), c_{\text{sell}}^{(i)}(k), c_{\text{prod}}^{(i)}(k), P_{\text{load}}^{(i)}(k), P_{\text{res}}^{(i)}(k)\right\}_k^{N_P}\right) \mid i = 1, \ldots, N_{sim} \right\}$$

3) Solve the optimisation problems using Gurobi [18].
4) Retrieve the binary decision tuples from the optimisation:

$$\Delta_N^j = \left\{ \left(\delta_1^{on(i)}(k), \ldots, \delta_{N_{gen}}^{on(i)}(k), \delta_{grid}^{(i)}(k), \delta_b^{(i)}(k)\right) \mid \forall i = k + j, \forall k = 1, \ldots, N_{sim}, j = 0, \ldots, N_p - 1 \right\}$$

5) Conduct data preprocessing to tailor the data for the transformer.

In our methodology, we utilise the microgrid data Z_N as source sequences for the transformer, with the corresponding binary decision tuples serving as target outputs. The source sequence consists of price and load sequences, and the state information is included repetitively over the prediction horizon to fit into the sequential format. It is assumed that the transformer can discern and weigh the relevance of the current state information throughout the sequence. This setup trains the transformer to predict operational decisions, reducing the need for iterative MILP solving. The following section details the specific data preprocessing steps required to facilitate effective learning by the transformer.

B. Data Preprocessing and Transformer Architecture

The preprocessing stage transforms raw data into a format suitable for the transformer, involving the normalisation of continuous features and encoding categorical variables to ensure uniformity and optimise the learning process. We employ sequence-to-sequence (seq2seq) architecture for processing and predicting binary decisions.

1) Source Feature Embedding: Inspired by vision transformers, which handle continuous image data by segmenting it into patches, we employ our source feature embedder for continuous operational parameters like energy prices and power demands [19]. This technique projects these inputs into a higher-dimensional vector space, standardising feature dimensions while preserving essential information. The transformation is defined as:

$$\mathbf{y} = W\mathbf{x} + \mathbf{b} \tag{6}$$

where $\mathbf{x}$ is the input feature vector, W is the weight matrix, and $\mathbf{b}$ is the bias. This method ensures dimensional consistency, allowing the transformer's attention mechanisms to process the data uniformly, thereby enhancing pattern recognition in microgrid management applications [20].

2) Binary Decision Processing: The binary decisions are encoded into integer values to simplify the target sequence for the transformer. This encoding facilitates the transformer's training by reducing the complexity of the output space.

3) Mini-Batch Processing: The training data are processed in mini-batches, enhancing computational efficiency and generalisation. This approach leverages batch normalisation to standardise inputs across each mini-batch, smoothing the learning process and stabilising the neural network training dynamics [20], [21].

C. Transformer Architecture

The transformer architecture employed in this study is specifically adapted to handle the unique characteristics of microgrid management data. Among the various components of the transformer, the source feature embedder stands out as a novel adaptation, enabling the transformer to effectively process continuous input data, a necessary departure from the typical usage of embedding in transformers in natural language processing:

- Positional Encoding: Adds temporal context to the inputs, compensating for the transformer's lack of inherent sequence recognition capability. Our data preprocessing enables a similar positional encoding technique as in [10], ensuring our data aligns seamlessly with the transformer architecture.
- Multi-Head Attention: Allows the transformer to focus on different parts of the input sequence simultaneously, enhancing the transformer's ability to capture complex dependencies and relationships within the data.

D. Transformer Training and Prediction

The transformer is trained using a cross-entropy loss function appropriate for classification tasks [22], with the Adam optimiser managing the optimisation of the transformer parameters. Training involves several epochs of processing data in batches, with validation to ensure the transformer generalises well to new data and monitor overfitting. Predictions are generated by processing input data through the trained transformer, converting the output sequences into actionable binary configurations using a predefined mapping, thus facilitating their integration into the MPC framework.

E. Cascaded Transformers

We introduce a cascaded transformer approach to address the infeasibility of predictions in transformer-based MPC, enhancing robustness and reliability. This method uses multiple transformers in sequence, beginning with a primary transformer that processes initial predictions and draws from a diverse dataset to manage a broad range of microgrid operations. If the primary transformer's output is infeasible, secondary transformers with different hyperparameters are activated to provide alternative predictions.

This setup ensures continuous adaptation and solution viability by sequentially engaging transformers until a feasible solution is found. It mitigates the limitations of single-transformer configurations, especially in complex and extended scenarios, by increasing system adaptability and fault tolerance. Further, in practice, secondary transformers can be tailored to specific operational scenarios, boosting system responsiveness and accuracy.

As shown in two example runs in Table II, introducing additional transformers significantly improves feasibility. While a single transformer achieves feasibility in 89.50% to 91.00% of cases, employing four transformers raises this to 98.10% and 98.40%, demonstrating the cascaded approach's effectiveness in maintaining operational feasibility across multiple runs.

TABLE I
FEASIBILITY CASCADED TRANSFORMERS

	Feasibility Run 1 (%)	**Feasiblity Run 2** (%)
1 transformer	89.50	91.00
2 transformers	95.20	96.50
3 transformers	96.50	96.80
4 transformers	98.10	98.40

F. Implementation Transformer-Based MPC

Predicted binary sequences are integrated into the MPC framework, simplifying the MILP problems into more manageable LP problems. By learning these sequences offline, we aim to reduce the online computational demands, enhancing the system's responsiveness and allowing for real-time operational adjustments in microgrid management. Figure 1 provides a schematic overview of the transformer implementation.

The methodology presented effectively leverages advanced machine learning techniques to improve the operational efficiency of MPC systems in microgrids, demonstrating the potential of transformers in energy control applications.

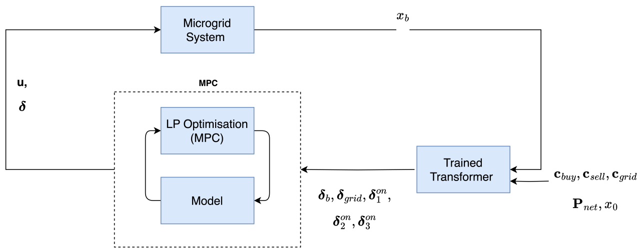

Fig. 1. Control Scheme Transformer-based MPC

VI. Simulations

This section presents detailed results from simulations conducted to evaluate the transformer for microgrid control using MPC. These simulations aimed to test the efficacy of transformers in forecasting binary decisions, which is crucial for reducing the computational load in MPC for microgrids.

A. Experimental Setup

The experimental setup aims to closely emulate typical microgrid operations to test the transformer under diverse conditions. The simulations utilised a MacBook Pro with an M1 Pro chip, running Python and PyTorch. Due to the lack of CUDA availability on this device, which is beneficial for enhancing transformer computations, all tests were conducted on the CPU.

B. System Model and Assumptions

The experimental foundation relies on the MLD system model detailed in section III, with the following assumptions made for simulation purposes:

- Predicted price and load data are fully known over the simulation horizon.
- The power generated by renewable sources within the microgrid is assumed to simplify the MPC problem. Our strategies remain relevant for scenarios using only predicted prices, loads, and renewable generation values, reflecting more realistic conditions.
- A consistent system configuration maintains a fixed number of generators.

These assumptions are pivotal in isolating the transformer's performance from external variability, focusing on the impact of hyperparameter configurations on its learning and generalisation.

C. Training Transformers

Training the transformer requires balancing hyperparameters for computational efficiency and prediction accuracy. We explored configurations with varying complexities. With its deeper layers and increased attention heads, the higher complexity setup effectively captured complex data relationships but was prone to overfitting and demanded more computing power. This led to longer training and prediction times, making it less applicable for real-time applications. Conversely, the lower complexity transformers performed adequately, especially for shorter prediction horizons, and are therefore preferable due to their reduced training time and quicker predictions, which are essential for real-time decision-making in microgrid management. In simulations, these transformers demonstrated an adequate balance between speed and accuracy, managing computational loads effectively.

D. Open Loop Simulations

To evaluate the transformer's effectiveness compared to other microgrid MPC strategies, we conducted open-loop simulations comparing three distinct approaches: a cascaded transformer, an LSTM-based MPC, and the optimal solution achieved using the Gurobi solver [18]. The prediction horizon of the MPC is set to 25, with a sampling time of $T_s = 30$min, which is equal to ≈ 12h. The results are summarised in Table II:

TABLE II
PERFORMANCE COMPARISON OF TRANSFORMERS TO LSTM AND OPTIMAL CONTROLLER

Metric	**LSTM**	Transformer	**Gurobi**
Training Time [s]	1796.84	891.88	-
Success Rate [%]	99.80	99.40	100
Optimisation Time [s]	0.0021	0.0487	0.0037
Optimality Gap [%]	0.18	0.76	-

In the open-loop simulation comparison, the LSTM-based MPC demonstrated superior performance over the transformer in terms of optimisation time and accuracy. While both transformers showed high success rates close to Gurobi's, the LSTM's optimality gap was narrower, indicating a closer approximation to Gurobi's optimal solutions. Though efficient in training, the transformer lagged in prediction speed, highlighting a critical bottleneck in its deployment for real-time applications.

This analysis underscores the need for further enhancements in the transformer approach to reduce prediction times and improve accuracy, making them more viable for energy management systems that demand rapid decision-making capabilities. Meanwhile, the LSTM's performance in this setting suggests it may be more suited for integration into such systems, where operational efficiency is critical.

E. Closed Loop Simulations

Following the open-loop analysis, we conducted closed-loop simulations to evaluate how the transformer-based MPC performs under dynamic conditions that mimic real-world operational feedback. These simulations adaptively updated the system's state, incorporating real-time system feedback and progressing iteratively through the microgrid data over time.

The closed-loop simulations were again conducted over a prediction horizon of $N_P = 25$, corresponding to approximately 12h, since each simulation cycle has a real-time step of $T_s = 30$min. We utilised a diverse set of initial conditions and external factors across 150 simulations to assess the robustness and reliability of the transformer-based MPC under varied operational scenarios. Performance was assessed based on average costs, computation times, and infeasibility rates. An infeasible configuration triggered a fallback to a standard optimal MPC strategy to ensure continuity. The results from these simulations are summarised in Table III.

TABLE III
COMPARISON OF CLOSED LOOP MPC SIMULATIONS

Metric	**Optimal MPC**	**Transformer-based MPC**
Avg. Costs	3786.50	3801.53
Avg. Computation Time [s]	31.38	87.33
Avg. Success Rate [%]	0	81

The transformer-based MPC generally displayed slightly underperformance compared to the optimal controller, with higher average costs and infeasibility rates. These results indicate the transformer sometimes struggles with dynamic state adjustments that are not well represented in the training data. Such scenarios reveal a need to enhance the diversity of the training datasets and possibly retrain the transformer to handle a broader range of operational dynamics more effectively.

This evaluation underscores the necessity for iterative development and testing of the transformer within closed-loop MPC. The insights gained from these closed-loop simulations will guide further refinements to improve the predictive accuracy and operational feasibility of transformer-based MPC systems in practical microgrid applications.

VII. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated the potential of using transformers to optimise microgrid MPC by accurately predicting binary decision sequences in MILP. We adapted transformer architecture to suit the characteristics of microgrid data and proposed methods to integrate cascaded transformers in the MPC framework effectively.

However, the real-time application of transformers in microgrid management presents substantial challenges. The extended computational times and increased infeasibility in closed-loop MPC experiments have underscored the need for further optimisation. These issues are partly attributed to our experimental setup, conducted using Python and CPU environments—configurations not optimised for high-speed, large-scale data processing.

Despite these limitations, our testing against traditional and LSTM-based MPC methods highlighted the accuracy and reliability of the transformer-based approach. However, the practical application of transformers is currently hindered by extended prediction times and the increased occurrence of infeasible solutions, especially with longer prediction horizons.

Future work should, therefore, focus on enhancing real-time capabilities by exploring more suitable computational environments, such as GPU-accelerated hardware, and refining the transformer models to handle the complexities of microgrid data better, particularly for longer-horizon MPC. Additionally, expanding the training datasets and further developing cascaded transformer strategies could improve transformers' reliability and accuracy.

These next steps are critical for advancing the integration of sophisticated machine learning strategies into the energy sector, aiming to significantly improve the operational efficiency and decision-making processes within microgrid predictive control systems.

References

[1] E. De Cian and I. Sue Wing, "Global energy demand in a warming climate," 2016.

[2] D. Gielen, F. Boshell, D. Saygin, M. D. Bazilian, N. Wagner, and R. Gorini, "The role of renewable energy in the global energy transformation," *Energy strategy reviews*, vol. 24, pp. 38–50, 2019.

[3] J. Rogelj, M. Den Elzen, N. Höhne, T. Fransen, H. Fekete, H. Winkler, R. Schaeffer, F. Sha, K. Riahi, and M. Meinshausen, "Paris agreement climate proposals need a boost to keep warming well below 2 c," *Nature*, vol. 534, no. 7609, pp. 631–639, 2016.

[4] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke *et al.*, "Trends in microgrid control," *IEEE Transactions on smart grid*, vol. 5, no. 4, pp. 1905–1919, 2014.

[5] C. F. Heuberger, I. Staffell, N. Shah, and N. Mac Dowell, "Levelised value of electricity-a systemic approach to technology valuation," in *Computer Aided Chemical Engineering*. Elsevier, 2016, vol. 38, pp. 721–726.

[6] T. Pippia, J. Sijs, and B. De Schutter, "A single-level rule-based model predictive control approach for energy management of grid-connected microgrids," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2364–2376, 2019.

[7] L. Fu, W. Zhang, Z. Dong, and K. Meng, "A mixed logical dynamical model for optimal energy scheduling in microgrids," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2017, pp. 1–6.

[8] D. Masti, T. Pippia, A. Bemporad, and B. De Schutter, "Learning approximate semi-explicit hybrid mpc with an application to microgrids," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5207–5212, 2020.

[9] A. Cauligi, A. Chakrabarty, S. Di Cairano, and R. Quirynen, "Prism: Recurrent neural networks and presolve methods for fast mixed-integer optimal control," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 34–46.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] B. van der Woude, "Github: Learning-based control of microgrids with transformers and model predictive control," https://github.com/bobvdw/Learning-Based-Control-of-Microgrids-with-Transformers-and-MPC-, 2024, gitHub repository.

[12] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[13] A. Parisio, E. Rikos, and L. Glielmo, "A model predictive control approach to microgrid operation optimization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1813–1827, 2014.

[14] S. R. Cominesi, M. Farina, L. Giulioni, B. Picasso, and R. Scattolini, "A two-layer stochastic model predictive control scheme for microgrids," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 1–13, 2017.

[15] P. Velarde, L. Valverde, J. M. Maestre, C. Ocampo-Martínez, and C. Bordons, "On the comparison of stochastic model predictive control strategies applied to a hydrogen-based microgrid," *Journal of Power Sources*, vol. 343, pp. 161–173, 2017.

[16] E. Rachelson, A. B. Abbes, and S. Diemer, "Combining mixed integer programming and supervised learning for fast re-planning," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2. IEEE, 2010, pp. 63–70.

[17] ENTSO-E, "The european network of transmission system operators: Transparency platform," https://transparency.entsoe.eu/, 2023, [Online; accessed 12-December-2023].

[18] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: https://www.gurobi.com

[19] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[21] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.

[22] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *International Conference on Machine Learning*. PMLR, 2023, pp. 23 803–23 828.

Bibliography

[1] E. De Cian and I. Sue Wing, “Global energy demand in a warming climate,” 2016.

[2] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, “State of the art in research on microgrids: A review,” *IEEE access*, vol. 3, pp. 890–925, 2015.

[3] D. Gielen, F. Boshell, D. Saygin, M. D. Bazilian, N. Wagner, and R. Gorini, “The role of renewable energy in the global energy transformation,” *Energy strategy reviews*, vol. 24, pp. 38–50, 2019.

[4] J. Rogelj, M. Den Elzen, N. Höhne, T. Fransen, H. Fekete, H. Winkler, R. Schaeffer, F. Sha, K. Riahi, and M. Meinshausen, “Paris agreement climate proposals need a boost to keep warming well below 2 c,” *Nature*, vol. 534, no. 7609, pp. 631–639, 2016.

[5] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, *et al.*, “Trends in microgrid control,” *IEEE Transactions on smart grid*, vol. 5, no. 4, pp. 1905–1919, 2014.

[6] C. F. Heuberger, I. Staffell, N. Shah, and N. Mac Dowell, “Levelised value of electricity-a systemic approach to technology valuation,” in *Computer Aided Chemical Engineering*, vol. 38, pp. 721–726, Elsevier, 2016.

[7] T. Pippia, J. Sijs, and B. De Schutter, “A single-level rule-based model predictive control approach for energy management of grid-connected microgrids,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2364–2376, 2019.

[8] L. Fu, W. Zhang, Z. Dong, and K. Meng, “A mixed logical dynamical model for optimal energy scheduling in microgrids,” in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–6, IEEE, 2017.

[9] A. Parisio, E. Rikos, and L. Glielmo, “A model predictive control approach to microgrid operation optimization,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1813–1827, 2014.

[10] D. Masti, T. Pippia, A. Bemporad, and B. De Schutter, "Learning approximate semi-explicit hybrid mpc with an application to microgrids," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5207–5212, 2020.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] E. Rachelson, A. B. Abbes, and S. Diemer, "Combining mixed integer programming and supervised learning for fast re-planning," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2, pp. 63–70, IEEE, 2010.

[13] B. van der Woude, "Github: Learning-based control of microgrids with transformers and model predictive control." https://github.com/bobvdw/Learning-Based-Control-of-Microgrids-with-Transformers-and-MPC-, 2024. GitHub repository.

[14] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[15] S. R. Cominesi, M. Farina, L. Giulioni, B. Picasso, and R. Scattolini, "A two-layer stochastic model predictive control scheme for microgrids," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 1–13, 2017.

[16] P. Velarde, L. Valverde, J. M. Maestre, C. Ocampo-Martínez, and C. Bordons, "On the comparison of stochastic model predictive control strategies applied to a hydrogen-based microgrid," *Journal of Power Sources*, vol. 343, pp. 161–173, 2017.

[17] D. Masti and A. Bemporad, "Learning Binary warm starts for multiparametric mixed-integer quadratic programming," in *2019 18th European Control Conference, ECC 2019*, 2019.

[18] A. Cauligi, A. Chakrabarty, S. Di Cairano, and R. Quirynen, "Prism: Recurrent neural networks and presolve methods for fast mixed-integer optimal control," in *Learning for Dynamics and Control Conference*, pp. 34–46, PMLR, 2022.

[19] J.-J. Zhu and G. Martius, "Fast non-parametric learning to accelerate mixed-integer programming for hybrid model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5239–5245, 2020.

[20] A. Richards and J. How, "Mixed-integer programming for control," in *Proceedings of the American Control Conference*, vol. 4, 2005.

[21] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021.

[22] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[24] D. Görges, "Relations between model predictive control and reinforcement learning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4920–4928, 2017.

[25] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.

[26] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[27] E. Bøhn, S. Gros, S. Moe, and T. A. Johansen, "Reinforcement learning of the prediction horizon in model predictive control," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 314–320, 2021.

[28] D. Zupanič, "Values suggestion in mixed integer programming by machine learning algorithm," *Electronic Notes in discrete Mathematics*, vol. 1, pp. 74–83, 1999.

[29] A. Cauligi, P. Culbertson, E. Schmerling, M. Schwager, B. Stellato, and M. Pavone, "Coco: Online mixed-integer control via supervised learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1447–1454, 2021.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[32] ENTSO-E, "The european network of transmission system operators: Transparency platform." https://transparency.entsoe.eu/, 2023. [Online; accessed 12-December-2023].

[33] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[35] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024.

[36] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.

[37] PyTorch, "Data loading and processing tutorial," 2023. Accessed: 2023-04-22.

[38] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *International Conference on Machine Learning*, pp. 23803–23828, PMLR, 2023.

[39] O. R. developers, "Onnx runtime." https://onnxruntime.ai/, 2021. Version: 1.14.0.

[40] Z. Zhang, Y. Chen, B. He, and Z. Zhang, "Niot: A novel inference optimization of transformers on modern cpus," *IEEE Transactions on Parallel and Distributed Systems*, 2023.

Glossary

List of Acronyms

LP	linear programming
LSTM	long short-term memory
MILP	mixed-integer linear programming
MLD	mixed logical dynamical
MPC	model predictive control
RL	reinforcement learning
SL	supverised learning