Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38
70569 Stuttgart

MasterThesis

Evaluating the Effectiveness of a Hybrid Approach for DSM in Unreliable Power Grids: Temporal Planning Meets LSTM

Usman Sikander Mirza

Course of Study: Information Technology

Examiner: Dr. Ilche Georgievski

Supervisor: Dr. Ilche Georgievski

Commenced: April 28, 2023

Completed: October 28, 2023

Abstract

The electricity demand-supply imbalance in numerous developing countries results in power outages, creating a need for innovative solutions. We categorize power outages into scheduled and unscheduled power outages. While extensive research exists on Demand-Side Management (DSM) in stable power grids, applying DSM strategies in unreliable power grids remains largely unexplored. Consequently, our study aims to implement DSM, combining AI planning and Machine Learning, to mitigate the impact of planned and unplanned power interruptions in smart homes within unstable power networks. Therefore, this research introduces a hybrid approach merging knowledge-driven based Temporal Planning with data-driven based LSTM model, utilizing the former to address DSM strategy together with scheduled outages, and the latter to proactively handle unscheduled outages. Furthermore, our study generally extends AI planning and especially the Temporal Planning by applying Timed Initial Fluents (TIFs), which have seen limited practical implementation to date. TIFs allow the declaration of numeric fluents at specific time points, expanding the expressive power for addressing complex problems. The implemented solution involves applying LSTM networks on household power dataset and weather information to predict unscheduled power interruptions. Leveraging the learned predicted model and Temporal AI planner, the system creates plans to ensure uninterrupted power supply and minimize energy costs using DSM strategies. The electricity cost reduction is performed by shifting the loads of the smart home to the battery during peak tariff hours, thus implementing DSM based-TOU technique. The cost of battery charging is optimized by charging it at the off-peak hours. Hence, the results obtained in this thesis indicate that our hybrid approach is effective for implementing DSM in unreliable power grids.

Contents

List of Figures

List of Tables

List of Listings

List of Acronyms

AI Artificial Intelligence

AI Temporal Planning

PDDL Planning Domain Definition Language

TIL Timed Initial Literal

TIF Timed Initial Fluent

POPF Partial Order Planning Forwards

POPF-TIF Partial Order Planning Forwards - Timed Initial Fluents

STRIPS Stanford Research Institute Problem Solver

IPC International Planning Competition

UBTMP Universal Bounded Trajectory Management Problem

BTMP Bounded Trajectory Management Problems

DSM Demand Side Management

TOU Time Of Use

DR Demand Response

ANN Artificial Neural Network

DL Deep Learning

RNN Recurrent Neural Network

LSTM Long short-term memory

ReLU (Rectified linear unit) Activation Function

SVR Support Vector Regression

MLR Multiple Linear Regression

ARIMA AutoRegressive Integrated Moving Average

GPSO Global best Particle Swarm Optimization

SVM Support Vector Machine

HRL Hierarchical Reinforcement Learning

MoDs Maps of Dynamics

TP True Positive

TN True Negative

FP False Positive

FN False Negative

HDF Hierarchical Data Format

NERC North American Electric Reliability Corporation

IESCO Islamabad Electric Supply Company

ENTSO-E European Network of Transmission System Operators for Electricity

PRECON Pakistan Residential Electricity Consumption Dataset

SAIDI System Average Interruption Duration Index

SAIFI System Average Interruption Frequency Index

CFL Compact Fluorescent Lights

SOC State Of Charge

LED Light Emitting Diode

AC Alternating Current

DC Direct Current

KWH Kilo Watt per Hour

KW Kilo Watt

REP Real Electricity Price

CPI Consumer Price Index

ELS Event-based Load Shedding

HVAC Heating, Ventilation and Air Conditioning

REST Representational State Transfer

API Application Programming Interface

SCADA Supervisory Control and Data Acquisition

ESS Energy Storage System

UK United Kingdom

USA United States

PWC PricewaterhouseCoopers

1 Introduction

Several developing countries have problems matching electricity demand with production. Due to the shortage of power production, utilities implement scheduled and sometimes unscheduled power outages as a power balancing mechanism. Khoury et al. states that systematic and frequent power outages are implemented in Nigeria, Lebanon, Egypt, Iraq, Pakistan, India [KMSM16]. High electrical demand, weather (high winds, storms, lightning) and equipment failures are leading causes of unplanned outages [Endb; Uni]. Most common power system faults are line tripping faults, that lead to unannounced load shedding [ZWLB17].

While extensive research has been conducted on scheduling device operations from peak hours to off-peak hours (DSM), there is a noticeable lack of studies addressing DSM in the context of unreliable power grids. The routine unreliability in the power grids of numerous developing countries motivates us to investigate and find solutions for implementing Demand-Side Management (DSM) under these unstable conditions.

Therefore, our problem statement is to apply DSM strategies from the consumer's perspective, in an unreliable power grids by exploiting Artificial Intelligence (AI) planning. Motivation is to find an amicable solution to provide interrupt free and low cost energy using DSM techniques to the attached devices in the smart home. We divide interruptions arising from the unreliable power grids into two types, planned and unplanned power interruptions. Also, this division is made by several electric utility companies to address this issue faced by them in unreliable power grids [MG23].

AI planning is domain-independent method, as the algorithms in AI planning or knowledge based models are not application-specific but generalized [Pla16]. The techniques we apply to solve a logistics problem can also be applied to a manufacturing problem, because the algorithms are independent from the subject area where it will be used. This is why AI planning is domain independent and we look for an approach that is methodical and objective. However, the problem with knowledge based models is that they are generic and static activity models [AALC15]. Since once they are defined, they cannot automatically evolve. That is if the environment changes, and the way activities or events are happened changes then it will not be able to provide fully accurate solutions. As a result, AI planning cannot address uncertainty related problems (e.g.,unscheduled power outages). These type of outages are random and can happen anytime.

In order to resolve the uncertainty while considering AI planning as well, we can apply data-driven based learning model that can learn and then predict those outages in-advance. Since these kinds of problems are best solved by machine learning algorithms. We researched related areas such as power grid fault detection and load prediction for addressing energy supply-demand disparities using machine learning methods. We found that machine learning algorithms work really well in power systems forecasting, and has become a significant method in comparison to other methods [KLWK20].

Now the question arises that how can we combine AI planning and machine learning? We propose a hybrid approach by utilizing both the knowledge-driven model and data-driven model. The data-driven techniques can sometimes demand large labeled data sets for specific parameters, and this make the model completely application specific or domain dependent [AA18]. This will be balanced by the AI planning model as it will solve the basic and standard cases (e.g., scheduled power interruptions and DSM strategy). Hence, the machine learning will only be needed to learn for unscheduled power outages. The hybrid approach helps us to combine the positive aspects of both the techniques.

Thus, in this thesis our main research question will be studying and investigating whether the hybrid approach is suitable for DSM in unreliable power grids. Also, probe and identify the factors which contribute to unreliable grid conditions and cause this randomness. This helps in creating the data-driven model effective. Further, we enhance the domain instance of AI planning to handle unscheduled power interruptions and take appropriate helpful actions. This enables the Temporal AI planner to operate proactively and take actions that cancels out the previously unpredictable adverse effects of an unreliable power grid.

As explained in the beginning of this chapter, we found that no research was conducted for applying DSM in unreliable power grids. More specifically, DSM was not applied earlier in a scenario which had scheduled and unscheduled power outages. Furthermore, data-driven based studies have mostly looked for load forecasting using weather variables. Apart from that, data-driven based fault detection studies have used real time measured frequency signals [WWT+19] and current, voltage along with active power data [ZWLB17] to forecast faults. These researches find out faults by just analyzing the measured power characteristics meticulously from the power network side, which is not applicable for us since we are modeling for a smart home that is located at the edge (consumer side) of the power grid network. Therefore, our research employs weather variables to develop a power outage detection model, addressing the gap in the application of DSM in unreliable power grids and the underexplored learning techniques for random power outage detection.

Further, during the literature review of machine learning in power systems, we found, there is large quantity of temporal information involved, extracting and capturing those hidden connections of features with temporal units is crucial [WWT+19]. LSTM deals better with longer time series and captures long-term temporal dependencies [ZWLB17]. Although, the exploration of data-driven fault prediction in power systems employing LSTM networks is in its early phases. LSTM model was used to timely detect and determine faults based on the available measurement signals [YWL16]. The findings from this study indicated that the LSTM network outperformed the ANNs when it came to the tasks of detecting and identifying faults in railway track circuits.

Therefore, we propose Long Short-Term Memory (LSTM) model to learn from the data-set and make predictions. As it incorporate the temporal attributes of input data, and has better performance in tackling time-series learning problem. LSTM is an improved Recurrent Neural Networks (RNN) [ZWLB17], RNN is an advanced form of Artificial Neural Networks (ANNs) [WWT+19].

The conventional PDDL has predicate logic, which can only assert a domain object true or false (e.g, turn ON lamp). The issue is that it was not possible to assert the predicates at specific times. Then, in PDDL 2.2 Timed Initial Literals (TILs) were introduced [EH03], they are the constructs that can assert the predicates true or false at specific time points (e.g., turn ON lamp at 7 PM). Further, Timed Initial Fluents (TIFs) are constructs that change the value of numeric fluents at some point in time (e.g., at 10 PM turn the intensity of lamp to 30%) [GSA23; IM]. So, TIFs provide huge expressive power for addressing complex problems. TIFs can offer contextual knowledge, for instance, the day-ahead hourly temperature. Since, our problem requires more expressive power because of its complexity so we use TIFs.

In context of the knowledge-driven part, we base our work upon Temporal planner POPF-TIF [PFL15] that is an extended version of Forward-Chaining Partial-Order Planning (POPF2) [CCFL15] and is capable of solving Timed Initial Fluents (TIFs). It supports numeric timed initial fluents or TIFs which is currently a state of the art feature in PDDL's temporal planning sub-domain. The application and practical use of TIFs in the context of studies related to Temporal Planning (TP) are quite scarce or not widely adapted to date. We could only find two published studies that have practically implemented TIFs to solve their particular problems. Piacentini et al. have applied TIFs to solve AC voltage control problems for power systems [PAFL15]. Georgievski et al. have demonstrated that their planning system significantly reduced energy costs compared to a baseline system, indicating the successful solution using TIFs to the problem of cost-effective energy management in buildings [GSA23]. Therefore, the contribution of our work in this particular application area constitutes the state-of-the-art in Artificial Intelligence together with Temporal planning domain.

We applied the LSTM model on a 16 months dataset of a house with data at an interval of every 5 minutes, along with that the weather for the whole duration was also fed into the model. Then the already learned predictor block predicts the random power interruption for the upcoming day by inserting the weather forecast via a weather application program interface (API). This is sent to the preprocessor which already has the load profile and the scheduled power interruptions, the preprocessor calculates the SOC decrement value for each hour for both scheduled and unscheduled power outages and express them as TIFs, then creates the problem file of the AI planning. Further, the AI planner creates a plan that provides uninterrupted power supply and tries to save electricity cost (DSM) by carrying out most of the charging actions during off-peak hours. The AI planner shifts the loads of the smart home to the cheaply charged battery during peak tariff hours (DSM based-TOU strategy), thus, solving our problem statement completely.

The 2nd chapter of this report offers foundational information and provides a straightforward explanation of essential terms. In Chapter 3, we review related work and provide our research contribution. Chapter 4 outlines the design of our proposed solution. Chapter 5 covers the implementation of our solution. Chapter 6 presents the results and their evaluation using various scenarios. Chapter 7 concludes with a final chapter dedicated to drawing conclusions.

2 Background Information

This section describes the concepts and terms relevant to this study, and help in understating the work.

2.1 Demand Side Management: Goal and Purpose

Demand-side management encompasses a wide range of activities aimed at reducing energy consumption and optimizing energy use. This can include simple measures like replacing old incandescent bulbs with more efficient compact fluorescent lights (CFLs), as well as complex approaches such as implementing dynamic load management systems, and more sophisticated approaches such as implementing dynamic pricing schemes for certain consumption patterns and using advanced control systems to manage energy use in real time [PD11]. The definition of DSM in the literature is broad, but the overall goal of DSM is to reduce the energy demand by shifting consumption patterns on the customer side [UMA+22].

Strbac defines DSM in terms of consumer's response to changes in price and shifting of load from on to off-peak periods [Str08]. It further adds that, in the past, having a capacity margin of about 20% was deemed enough to ensure adequate safety of power generation. This was based on the average demand throughout the year, which typically results in an average utilization of the generation capacity of less than 55%. This means that there is a considerable opportunity for Demand Side Management (DSM) to shift the load from peak to off-peak periods, which would reduce the requirement for additional generation capacity and increase the utilization of existing generating plants. As a result, this can help to reduce the need for new power plants and other expensive infrastructure. Also, DSM assists in solving the supply-demand gap so it can improve the reliability of the energy system by reducing peak demand and avoiding overloads and blackouts.

Strbac predicts that in future DSM can support the integration of renewable energy sources, such as wind and solar, by helping to balance supply and demand and reduce the need for burning huge amounts of fuel [Str08]. The system will need to apply large amounts of reserve, to absorb energy by renewables when demand is less.

Essentially, DSM refers to any actions taken on the demand side (consumption side) of the energy system to improve efficiency and reduce energy use that in turn reduce greenhouse gas emissions. The goal of DSM is to reduce energy demand, lower costs, and improve the efficiency and sustainability of the energy system as a whole.

2.1.1 Categories of DSM

Palensky et al. states that DSM can be categorized into four types [PD11]. The categorization depends upon the timing and impact of applied measures on the customer side. In the figure shown in study, the types that are close to the origin are permanent changes on equipment and highly optimized. While, schemes that are away from the origin are reduced briefly and have less optimized scheduling.

- Energy Efficiency (EE): Permanent changes like exchanging an inefficient ventilation system with a better one, or adding an extra insulation layer on the building to conserve energy.
- Time of Use (TOU): Time of use tariffs penalize certain periods of time (e.g., 17:00–19:00) with a higher price, so customers (re)arrange their processes to minimize costs.
- Demand Response (DR): Demand response (DR) provides the platform for end-users to alter load consumption patterns in response to a price change of electricity over a period, thereby reducing the overall peak of the system [UMA+22].
- Spinning Reserve (SR): On the demand side, this means that load can be reduced or increased when the grid frequency drops or rises, this is done by the power plants. In figure, they are on the upper (quick) end of the DSM spectrum.

An alternate way to look at the flavors of DSM is Market DR and Physical DR [PD11]. In market DR, it has real-time pricing, price signals and incentives. When the incentive based market DR does not work alone completely, then grid managers move to physical DR. In Physical DR, grid management is done, and emergency signals are sent for DSM if the grid or parts of its infrastructure are in a low performance state due to some fault or routing maintenance.

Albadi et al. provides a different view for the classification of DSM [AE07]. It terms that one is DR and the other is EE. They define DR as it provides the platform for end-users to alter load consumption patterns in response to a price change of electricity over a period, thereby reducing the overall peak of the system. While, EE emphasizes persuading consumers to utilize efficient products as a means of reducing demand.

Important to note is that only the EE reduces consumption, all other methods just shift the time when that energy is to be used. They shed the load or shift it.

2.1.2 Time of Use

Price-based programs are employed to level out the demand curve by offering elevated prices during peak periods and reduced prices during off-peak periods [AE07]. In this way, customers are incentivized to shift their energy usage to off-peak hours, which can help to reduce the strain on the grid during high demand periods.

Utilities use time-of-usage (TOU) based electricity pricing to set different rates at various times of the day as the load demand fluctuates [DMB22]. Within the micro-grid system's hourly demand, two types of loads are present: elastic and inelastic. By reorganizing the demand model based on demand-price elasticity through demand-side management (DSM), elastic loads can be shifted from peak load hours to lower-cost hours. When implementing the DSM strategy, computed numerical results have demonstrated significant savings of 8% to 18% across all test systems [DMB22].

Palensky et al. states that TOU is a good policy [PD11]. Further, DSM & TOU is highly beneficial as it lowers costs. TOU can be mapped hierarchical using the figure provided in [UMA+22]. DSM to DR to Price-Based Program to TOU, hence, it is a price based DSM strategy.

For this study, we consider TOU approach to achieve DSM in our model. We shift the load of residential unit from peak times to off-peak times. For shifting the load a slightly different method is used, the load is shifted by using a backup resource using peak time and charging it during off-peak time.

2.1.3 TOU Use in Different Countries and Pricing

The TOU method is widely practiced in European countries [Str08]. Similarly, Usman et al. states that in both the United States and Australia, numerical trials have been conducted to evaluate the effectiveness of peak load reduction through the use of price-based programs such as real-time pricing, time-of-use pricing, and critical-peak pricing [UMA+22] . These programs were primarily implemented across the residential, commercial, and industrial sectors of their energy systems, and were found to enhance the security of energy supply.

TOU based DSM strategy has benefited numerous countries and would be beneficial for Nigeria as it transitions to the medium-term energy market [UMA+22]. DSM strategies like peak clipping and valley filling are employed in price-based programs such as Time-of-Use (TOU) and Real-Time-Pricing (RTP) to curtail power demand during on-peak periods and augment demand during off-peak periods.

By offering customers the opportunity to save while shifting energy use to off-peak periods, time-of-use programs like Octopus Agile, Octopus Go, and Go Tariff have played a crucial role in energy management [Hom]. The peak hours in the UK are from 4 pm to 7 pm during November to February.

Islamabad Electric Supply Company (IESCO) provides two different prices per Kilo Watt Hour (KWH) of electricity usage according to their TOU program [Isld]. For peak usage they charge 34.39 rupees per KWH, and for off-peak they charge 28.07 rupees per KWH. The peak timing for the complete year are pre-calculated according to power grid demand-supply condition and sun rise/fall times, and defined for each quarter, from Dec to Feb peak time is 5 pm to 9 pm, from Mar to May peak time is 6 pm to 10 pm, from June to Aug peak time is 7 pm to 11 pm, from Sep to Nov peak time is 6 pm to 10 pm. The remaining 20 hours of the day are considered as off-peak timing respectively.

2.1.4 Overview of DSM and TOU in Developed Opposed to Developing Countries

The section number seven provides a comparison of DSM programs in developed and developing countries [UMA+22]. The developed countries (US, UK, Australia) use almost all DSM programmes, while developing countries (India, South Africa, Nigeria) use Price based DSM model & Energy efficiency improvement model more, in comparison to developed countries. The developing countries do not use the incentive-based model, but developed countries do implement them.

This assessment is summarized from table number ten of [UMA+22], in which DSM programs such as, Price based (example: TOU), Incentive based (example: direct load control) and Energy efficiency improvement (example: LED) are shown perpendicular to the developed/developing countries and their implementation given as majorly implemented, partially implement and not implemented is presented.

Hence, the comparison provides interesting details and a basis for this study to use Price-based DSM model (example: time-of-day/time-of-use pricing), because developing countries generally implement this model more frequently in comparison to other models.

2.2 Artificial Intelligence Planning

Georgievski et al. explains that AI planning is a powerful tool that allows us to define and solve complex planning problems [GA16]. At the core of AI planning lays a set of key concepts that include the initial state, the target state, and a set of actions. The initial state represents the environment before any plan execution, and the target state defines the desired state of the environment after the plan has been executed. While actions consist of parameters, conditions, and effects.

To carry out a successful action, the parameters must be filled with appropriate values. Preconditions, which contain predicates that must be satisfied for an action to be carried out, and effects, which contain predicates that simulate the action, are also crucial. Predicates can either have a true or false value and are composed of names and parameters that describe the relationships between them.

Executing an action transitions the initial state to a new state, which is achieved through the effects. A plan is essentially a series of actions that, when applied to the initial state, leads to the defined goal. Once a plan has been executed and it achieves the desired outcome, it is considered a solution to the planning problem. Overall, AI planning provides a systematic approach to solving complex problems and is a valuable tool for a wide range of applications.

2.3 Planning Domain Description Language

To model and practically implement AI planning in a more standardized way PDDL is used now. In 1998 Planning Domain Description Language (PDDL) was defined [MGH+98]. It made research in AI Planning more modular and easily comparable.

PDDL places a strong emphasis on actions and is influenced by the Stanford Research Institute Problem Solver (STRIPS) [FN71]. The language is designed to formulate planning problems using a standardized syntax that describes the semantics of actions used in STRIPS, with pre- and post-conditions defining the applicability and effects of actions.

It was decided at the start to separate parameterized actions, which represent the domain behavior. The description of the objects, the initial state, and the defined goal are represented in the problem instance. This separation allows for a more modular and flexible approach to problem-solving, and enables the development of realistic applications in a variety of domains. Hence, an AI planning problem that is being modeled in PDDL requires two separate files, domain and a problem file [M

F03]. PDDL has been widely adopted as the basis for the scientific advancement of planning, and has proven to be a useful and expressive tool for solving complex problems. In response to the growing demand for more advanced planning and scheduling techniques, PDDL has been further developed and modified to tackle even more challenging tasks, making it a helpful resource for autonomous planning and scheduling applications.

2.4 Temporal Planning

In classical planning, the simplest form of planning is to have actions performed in a one by one format, sequentially (one after the other). When several things start happening at a time, it is required to model the duration and concurrency of actions. Multiple actions can be executed simultaneously with varying durations, although actions and events have complex interdependencies that determine which actions could be started along with others. Rintanen describes this as Temporal Planning [Jus23]. It also clarifies that scheduling problems are different in comparison to temporal problems, because the tasks are already given, only they have to be scheduled (assigned a starting time). While in the latter, the scheduling of the action/task is the same except that the choice of actions is left open, and they must be selected along with their scheduling. So there is a choice of actions/tasks from which an optimal action has to be selected at an appropriate and best starting time.

Haslum et al. explains that when classical planning is extended in 2-dimensions, with the second dimension as time then the result is Temporal Planning [HLMM]. In temporal planning, actions are durative in nature so they take time to execute, which could be defined in PDDL version 2.1 [M F03]. PDDL version 2.1 was introduced to support temporal planning domains. It was extended from the PDDL version 1.2, this added durative-actions and functions (that are numeric fluents). Numeric fluents (numbers) are just like variables in other programming languages that can store numeric values and certain mathematical operations can be performed on them. They allow us to express resources, temperature or battery charge etc. Like predicates in PDDL, numeric fluents can be put as a condition in an action, and also as an effect of an action, for example in modeling the consumption of a resource. Numeric fluents are defined using the keyword “function” in PDDL.

Georgievski et al. suggests that temporal planning is appropriate for solving the so-called building coordination problem, because such problems need strong representation of numbers and time in them [IM]. And temporal planning allows modeling problems like these which need numeric and temporal constraints. Thus, it aids in the main objective of building automation, which is to increase safety, improve people’s comfort and productivity, increase energy efficiency and reduce operational costs. In our study we apply and implement temporal planning onto a residential unit to improve people’s comfort (by providing uninterrupted power in an unreliable grid setup), and increase energy efficiency (intelligently charging battery at off-peak rate and discharging during peak time) that ultimately reduce operational costs.

2.4.1 Durative Actions

They have pre-conditions, effects and duration constraints. Hence, actions occur over a time span, they have an execution time (length of time an action takes). According to Fox et al. the preconditions can be checked at start, or at end or during the whole execution (with the keyword "overall") [M F03]. Also, the effects can be applied at the beginning or end of the action, which will change the domain functions accordingly.

Duration can be represented as anything in real world time, for example; duration of 1.0 can be modeled to represent 1 minute or 1 hour etc. For the sake of simplicity, in this study we use 1.0 as 1 hour. As we generate day ahead plans, when the duration of 24 has reached (or 24 hours) then the day has ended. It is not possible to assign a numeric fluent to the duration. Thus, the duration cannot be referred to as a numeric fluent (variable) in the respective action, it must be defined as a numeric value.

2.4.2 Timed Initial Literals

Timed Initial Literals were newly introduced into the PDDL during the 4th International Planning Competition (IPC-4) in the year 2004 [EH03]. The language for the classical part of the IPC-4 was called PDDL version 2.2. Derived predicates were re-introduced and timed initial literals were first introduced in it; these two constructs were the main advancement of PDDL version 2.2. The syntax and semantics of these constructs were defined.

TILs are constructs that indicate to us that at what point in time a fact becomes true which was previously false [IM]. They make the predicates as true or false at specific time points. Also, TILs can be utilized to model predicted changes that may occur over the course of the building operation. By specifying TILs, we can track the evolution of different predicates and conditions over time, allowing us to make more accurate predictions about the behavior of the system. For instance, we can specify them as, e.g. the sun rose up at 7 am (at 7 (sun-rise)), so open up the curtains.

2.4.3 Timed Initial Fluents

In addition to TILs, Timed Initial Fluents (TIFs) can also be used to represent temporal constraints, and additionally they have numeric values over specific time intervals [IM; PAFL15]. Although TIFs are not yet officially part of the PDDL specification, they can be utilized by planners that support them [Baj16].

Bajada also gives an example for a TIF; (at 10 (= (temperature) 18)), it means that at 10 am the temperature is 18 degree centigrade [Baj16]. Hence, we can use TIFs to represent the temperature at different times of the day. This allows us to reason about the relationship between time, temperature, and other relevant parameters in the system, enabling us to make more informed decisions about resource allocation, energy management, and maintenance. By incorporating TIFs into our modeling and planning frameworks, we can create more comprehensive and accurate representations of the system's behavior, which can help us optimize its operation and management.

Certain components in a building's operation may have flexible or variable durations, which can be controlled on-the-fly by the building automation system [IM]. For instance, operations involving batteries, such as charging, may have variable durations that can be adjusted to meet the changing needs of the system. In contrast, uncontrollable components provide background information that is temporal in nature and cannot be directly controlled. For example, the day-ahead market provides hourly electricity prices, which can be used to inform the building automation system about when to perform energy-intensive tasks and when to conserve energy.

Similarly, in this study we have used the TOU pricing (uncontrollable component) as a background information that is temporal in nature and cannot be controlled. Afterwards, we schedule the different battery charging actions at different times (controllable component) to meet the changing needs of the residential unit and reduce electricity cost.

2.5 AI Planner

Apart from the PDDL which defines the planning problem, the AI planner is another crucial component in the process of AI planning. And selecting an AI planner is an important task. The AI planner is capable of reading and breaking down the PDDL and then attempting to find a solution for the problem.

Numerous types of AI planners in great numbers are available and new ones are also periodically introduced to cater to specific and novel use cases [A G]. Although, depending upon the type of planning problem a planner is selected. Since the problem in this study is related to TP, the planner must be able to support temporal constraints, such as durative actions, TILs and TIFs. We selected POPF-TIF as the AI planner for our specific problem as it is able to solve problems that have temporal constraints and most importantly TIF solving capability. POPF-TIF is basically based on POPF2 [PAFL15; PFL15].

The other planner that can solve using TIFs is the planner UPMurphi [PMM09] that was presented in the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009). Piotrowski et al. explains that UPMurphi can handle TILs and numeric TIFs or simply TIFs [PFL+]. The programming syntax of UPMurphi is slightly different from the standard PDDL, although for POPF-TIF it is the same. First and important reason to selected POPF-TIF was that it uses standard PDDL language, secondly, Piacentini et al. has previously solved a problem which is somewhat related to the one we are solving [PAFL15]. So it provides a foundation, and already a test case has been solved using planner POPF-TIF.

2.6 Advanced Temporal Planning-based Problems

In this section we define a new type of problem that can be appropriately solved with AI Planning, more specifically using Temporal Planning. Piacentini et al. introduced a new class of metric temporal problems, which are described by numeric background events and a trajectory constraint [PAFL15].

Further, they defines some terminologies.

- A numeric fluent that is represented in TIFs and remains unchanged by any action effects is referred to as an uncontrollable fluent. Meaning the domain file has no influence over it and it has to be provided via the problem file.
- A numeric fluent that is affected by action effects but remains unaffected by Timed Initial Fluents is known as a controllable fluent. A controllable fluent is affected and modified by the domain file.
- A term that has both controllable and uncontrollable fluent is called a mixed metric expression.
- A comparison between a mixed metric expression and a constant value.
- A Universal Bounded Trajectory Management Problem (UBTMP) is a planning problem in which all numeric fluents are either controllable or uncontrollable and it has a single trajectory constraint.

Piacentini et al. have defines Bounded Trajectory Management Problems (BTMPs) as a new class of problems and adds AC voltage control as an example of this class [PAFL15]. BTMPs refer to problems that involve managing and maintaining a certain quantity or parameter within safe or desired bounds. Some examples of BTMPs include controlling the speed of a vehicle, regulating the temperature of a room or building, and managing network voltage levels. These problems can also be viewed as control problems, such as using a thermostat to control heating and cooling systems to maintain a comfortable room temperature. These problems are particularly interesting from a planning perspective when the control actions take a significant amount of time to respond (durative actions) to changes in the background processes.

2.6.1 Envelope Action

The integration of temporal planning, TIFs, and trajectory constraints poses a significant challenge that existing planners are unable to overcome. Nonetheless, it is feasible to transform the "always" constraint and TIFs into features that can be handled by certain temporal planners [PAFL15].

The "always" is a construct for implementing trajectory constraints. Although, PDDL3 language [GHL+09] supports the specification of trajectory constraints such as (always P), (sometimes P) and (hold-after t P) and many more, but no earlier version of PDDL does not support these constructs. And to use the constructs such as durative actions, TILs and TIFs we have to use PDDL2.2. Hence, the work done in [PAFL15] elegantly models the "always" constraints into PDDL2.2 using a so-called "envelope action".

So according to Piacentini et al. an "envelope action" refers to an action that initiates before any other action in a plan, and concludes only after all other actions have been executed [PAFL15]. To ensure that the envelope action commences at the start of the plan, it needs to have an at-start precondition that specifies a proposition provided in the initial state and deleted by a TIL after some time (or just after envelope action starts). This allows sufficient time for the envelope to initiate before the condition is removed.

Moreover, the envelope action should have a start effect that confirms a proposition, which is specified as a precondition for every other action, sort of like a go-ahead or unlocking condition for other actions/durative actions. This ensures that no other actions start before the envelope opens. Lastly, the completion of the envelope action should not occur until the problem's goal is accomplished. This can either be at the specified time horizon or when a certain condition is met, which corresponds to the achievement of the goal.

The planner can create day-ahead plans, it starts when the envelope opens and stops when the envelope ends, and in-between the envelope all other important durative actions can be performed. When it has avoided the anticipated violation (given as TIF) by applying an helpful action then it keeps on going and caters for the next TIF, and does not reach the final state. Contrary to the working of action / durative action in PDDL 2.2 or earlier versions, in which if the actions have run even once and the goal conditions are satisfied then it reaches the end state. But in our problem, we want the actions to keep happening, adjusting the metric expression, to fulfill the boundary constraint and the planner to keep running until the end of the day even if it has no action to perform (in case it remains within predefined confines when no violation was anticipated in the future). Because the goal is to keep it running for 24 hours and during all this duration the metric constraint must be satisfied. When 24 hours have passed and the battery's SOC has remained in bounds then the stop condition is asserted and the envelope action stops. This is how the system works for 24 hours a day.

2.6.2 Enhancing Planning with TIF based Heuristics

As discussed above TIFs give us background event information that is fed into the problem file. The Figure 2.1 below shows a portion from the problem file of a simple planning problem in which TIFs are shown at different times. The battery's SOC is decreased because of an external source. The planner gets this data in the form of TIFs and then acts accordingly.

```
(at 16.0 (= (battery_soc) 70))
(at 17.0 (= (battery_soc) 70))

(at 18.0 (= (battery_soc) 63.95))
(at 19.0 (= (battery_soc) 53.47))
(at 20.0 (= (battery_soc) 34.68))
(at 21.0 (= (battery_soc) 21.19))
(at 22.0 (= (battery_soc) 10.36))
```

Figure 2.1: TIFs in a problem file

To understand the use of envelope action and TIF, consider an example in which a battery system can be charged and discharged. The domain file is such that it has an envelope action that has to keep the value of SOC between 40%-100%. We have introduced two types of other durative actions. Those are "chargeBattery" and "dischargeBattery". In the "chargeBattery" action the battery is charged for 1 time unit and at the end of the action 10% SOC is added as an effect. While in the "dischargeBattery" action the battery gets discharged in 1 time unit and at the end of the action 10% SOC is subtracted as an effect. The listing of the code from the domain file is shown in the Figure 2.2 below. We will explain the various pre-conditions and effects in later sections, this example is just for conceptual undertaking.

```
(:durative-action chargeBattery
:parameters()
:duration (=?duration 1)
:condition(and
            (at start(enable))
            (at start (is-not-decreasing))
)
:effect(and
(at end (increase (battery-soc-fix) 10))
(at start (not(is-not-increasing)))
(at end (is_increasing))

))

(:durative-action dischargeBattery
:parameters()
:duration (=?duration 0.5)
:condition(and
          (at start(enable))
          (at start (is-not-increasing))
)
:effect(and
(at end(decrease (battery-soc-fix) 10))
(at start (not(is-not-decreasing)))
(at end (is_decreasing))
))
```

Figure 2.2: Sample durative actions

Since the battery SOC reduces considerably at the TIFs presented in the problem file (Figure 2.1), e-g: at 18, at 19, at 20, at 21 and at 22, so the planner needs to apply the helpful actions in order to avoid the anticipated violations. At time unit 20, 21 and 22 the value of the SOC crosses below the limit of 40% and is thus reduced significantly. Therefore, the planner intelligently finds out that the "chargeBattery" action is helpful as it adds to the SOC and helps in maintaining the SOC between the bounds. Thus it is applied at times 19.671, 20.671 and 21.671 as shown in Figure 2.3. Thus, the bounded problem is solved using this approach.

```
0.000: (dayaheadplan)  [24.001]
19.671: (chargebattery)  [0.330]
20.671: (chargebattery)  [0.330]
21.671: (chargebattery)  [0.330]
Planner found 1 plan(s) in 0.074secs.
```

Figure 2.3: Output of AI planner

Had the case been opposite, that in some other case the TIFs gave information that battery's SOC had increased to over the limit, then POPF-TIF planner would have applied "dischargeBatttery" action. For instance, assume a certain temperature range is to be maintained by a heating ventilation and air conditioning (HVAC) system, and we have two actions, cooling and heating. If for example the temperature of the room goes above the range then cooling action would be taken as its effect would reduce the temperature. And if temperature goes down, then the heating action would be applied as its effect would increase the temperature in the room. Thus, the approach presented has the effect of improving the heuristic value by considering the actions necessary to fulfill all conditions, while also ensuring that helpful actions are included in the plan to achieve these objectives [PAFL15]. This is critical for the planner to avoid violations caused by TIFs, which could occur if these actions were not taken into account. Without this, the plan would not include these useful actions, leading to incorrect decisions by the planner.

Further, the mechanism can be thought of as a look-ahead that anticipates future TIFs and allows the heuristic to identify actions that can help avoid any violations they might cause. The authors have limited the look-ahead to the next TIF relevant to that variable, which they say is a cautious approach since the plan must respond to this TIF before it occurs.

2.7 Machine Learning

Machine Learning (ML) is defined as a field of artificial intelligence that focuses on developing algorithms and models that enable computers to learn from and make predictions or decisions based on data [Mit97]. Another meaning of ML in terms it is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment [NLM15]. We have used ML in this work to learn from a large data-set and then make predictions for the random loss of power supply.

2.8 Unreliability

North American Electric Reliability Corporation's (NERC) state of reliability report for the year 2022 defines the reliability of interconnected bulk power systems in terms of three basic and functional aspects [NER].

- Adequacy: That is the ability to meet the power needs of consumers continuously, considering both scheduled and unscheduled outages.
- Operating Reliability: It means that the ability of an electric system to withstand sudden disturbances, such as electric short circuits and unanticipated loss of system components.
- Adequate Level of Reliability: The design, planning and operation of the basic electric system will be achieved when it does not experience cascading and voltage collapse, or when frequency and/or voltage is kept within predefined parameters under normal operating conditions when subject to predefined disturbances.

These were the conditions which are required for a power producer or a grid system to be called reliable. Unfortunately, power grids of many developing countries are unable to meet the conditions listed above, and they fall into the category of unreliable grid conditions. According to the "Adequacy" bullet point above it can be said that a reliable power grid is the one which does not have scheduled or unscheduled power outages. Most common faults in a power system are line trip faults, which lead to un-announced load shedding or random loss of power, it can often trigger huge scale blackouts in the electricity grids [ZWLB17].

Further, adds that system operators should introduce counter measures to maintain a balance between supply and demand [NER]. When there is a shortage of capacity, they must take steps to remedy the situation. That includes, interruptible demand for some users using curtailment agreements, voltage reductions (also called brownouts), and rotating interruptions/outages. In rotating outages an already known set of distribution stations are interrupted for some limited time, then put back on and another set is cut-off, hence, it is called rotating outages. Rotating outages are also called blackouts or power outages or load shedding or force shedding. In this thesis, as a matter of choice we use the terms power outages and load shedding frequently in lieu of the term rotating outages. While referring to a few citations in its introduction Li et al. states that load shedding (LS) control is an effective countermeasure against voltage instability (which is a type of unreliability) [LXR20].

2.8.1 Scheduled Power Outage Standard

NERC defines a standard for power outages / load shedding that is titled as Load Shedding Plans, standard EOP-003-1 of the NERC [Nor]. It is a standard procedure to shed customer load when the transmission operator or balancing authority operates with insufficient generation or transmission capacity, this is done to avoid risk of component failure and cascading outages. They do it because of under-frequency and/or under-voltage conditions. The standard also specifies that each balancing authority / utility gives its consumers their currently enforced load shedding plans.

2.8.2 Scheduled and Unscheduled Power Outages

The portal [Strb] publishes electricity related information, renewable energy sources and legal regulations in Macedonia particularly and European Union in general. It also notifies about upcoming power interruptions with times and area / localities, for instance, the interruptions for 18 October, 2023 are published [Stra]. Most of the routine interruptions are to balance demand with available supply and these power outages last for 1 hour, could be more in some areas. For example, the first two interruptions in the city of Veles, Macedonia are for 1 hour each, and in other cities or villages even more. But these are scheduled power interruptions since intimation for these outages are done in advance. Also, the Skopje Electricity Distributor publishes a list of planned power outages on their website in the “Planned Disconnections” tab.

The “Strategy for Energy Development of the Republic of North Macedonia until 2040” depicts a prominent finding that the duration and frequency of electricity supply interruptions in distribution network in North Macedonia are relatively high compared to region [Pri]. This can be seen in the figure number 1.32 in the mentioned report, it shows planned along with unplanned System Average Interruption Frequency Index (SAIFI) and System Average Interruption Duration Index (SAIDI) indicator for distribution (excluding extremer weather conditions) in 2016 for counties in the region.

Astal et al. outlines key factors influencing electricity demand and reliability, including disruptive weather, which disturbs the electricity supply chain and leads to energy usage fluctuations and decreased plant efficiency, affecting the supply [AAAF22]. If the weather variation is sudden then this can lead to unscheduled power outages. Similarly, in the summers during September 2011, South Korea experienced a serious demand and supply imbalance which led to nationwide scheduled power outages [SK20]. Though, it did not happen a month before as humid weather continued and heavy rainfalls replaced the heat wave. To sum it up, we say that whenever there is an issue in demand-supply then power outage occurs somewhere in the electricity network.

Islamabad Electric Supply Company (IESCO) of Pakistan that is a local (balancing authority in our case) is responsible for providing electricity in the Pakistani capital. It coordinates its load shedding plans with her consumers by its website. It provides a load shedding plan for one complete month in advance [Islc], and also provides a monthly maintenance schedule in advance [Islb]. The planned power outage file can be downloaded in an excel file format, that include that grid name, feeder name/area and times on which the electricity would be cut-off and the duration of cut-off, usually the outages last for one hour.

The power interruptions in developing countries are mostly planned by the local electricity supply companies, like IESCO, but unplanned interruptions do occur. IESCO issues a quarterly report in which it provides the number of planned interruptions, unplanned consumer power supply interruptions, incidents of voltage fluctuations and frequency fluctuations etc. In this quarterly report of 2021, the no hours electricity was cut-off during planned interruptions and unplanned are presented [Isla]. It had 541 hours of total planned interruptions while 67 hours of unplanned interruptions for all feeders of the utility in 3 months. Hence, both the planned and unplanned interruptions highly impact in making a power grid unreliable, this gives us the motivation to investigate and solve this problem from the consumer's end by simulating a smart home.

Endeavour Energy provides electricity in some parts of Australia, they have a real-time power outage map. The map displays real-time information for both unplanned and planned outages along with their locations [Enda].

2.8.3 Problems in grids of developing countries

The IESCO report states that the reason for planned interruptions is the gap in demand and supply [Isla]. India ranks as the world's third-largest energy consumer and possesses the fifth-largest electricity grid, as well as the third-largest distribution and transmission network, boasting a total installed capacity of 372,693 MW of electricity [Intb; Min]. However, the country's electricity network is confronted with various challenges, including voltage fluctuations, recurring load shedding, inadequate support for reactive power, and suboptimal metering systems [Sub]. Also, the Indian grid experiences imbalance in frequency because of variable renewable energy generation [UMA+22]. The share of renewable energy in the Indian grid is 19.79% of the total installed capacity, and the energy mix varies 6% to 10% over a month which when fed into the grid creates problems. The lack of an adequate storage system (battery) with renewables creates this issue.

South Africa is a developing country with an installed capacity of 48,000 MW of electricity [Tho], for a large population of 59,978,099 [Wor21]. Significant power shortages have arisen due to the depletion of the electricity reserve margin, necessitating the use of load shedding as a balancing mechanism [Mon17; MVC17].

In our study, we consider all types of unreliability conditions such are, demand supply gap, voltage fluctuation, frequency fluctuation, reactive power issue etc., as complete loss of power, which is called a power outage. This is done because we are designing a system for a residential unit and also it goes out of scope to regulate different parameters of the power grid.

2.8.4 Reasons for problems in grids of developing countries

In this section we examine the reasons and causes of unreliability in the power grids. Insufficient gas or fuel supply, limited distribution network, difficulties in managing water resources, inadequate capacity of transmission lines, and other related infrastructure issues have contributed to the decline of the power sector in Nigeria [Nig21].

The lack of funding for power stations, ineffective communication infrastructure, delayed maintenance of facilities, low staff morale, outdated equipment and tools, safety concerns, and a shortage of operational vehicles are also factors contributing to the decline of the Nigerian power grid [SGZG10], that makes it an unreliable power grid.

The most frequent causes of line trip faults are weather variations, poor insulation, aging and damage to the distribution equipment [ZWLB17]. Astal et al. states lists prominent factors that influence electricity demand, and thus cause unreliability [AAAF22]. The factors are, disruptive weather that disrupts electricity supply chain, including electricity generation, transmission and distribution. It is because the drop in temperature and humidity lead to spikes and drops in energy use. It also observes that the efficiency of the plant producing the electricity also decreases, which ultimately drops the supply. In the same way, weather change affected South Korean power grid to experience demand and supply disparity [SK20]. Hence, weather plays a large role in causing problems to power systems.

Common causes of unplanned power outages nationally and locally are weather (high winds, storms, lightning), high electrical demand and equipment failures [Uni]. Endeavour Energy, provide of electricity in some parts of Australia states that High winds, lightning, storms, bushfires, car accidents, trees and branches falling onto power-lines or equipment failure can disrupt your power without warning are causes of unplanned outages[Endb]. Hence,extreme weather conditions or very high demand highly impact the power grid network.

PricewaterhouseCoopers(PwC) gives the recommendation for Macedonia, that the SAIFI and SAIDI indicators can be improved by investing in the distribution network to improve the supply reliability. The major factors for these investments are distributors' investment capacities, amount of approved investments by the regulator and the role of state institutions during the development and construction phase of infrastructure [Pri].

Lack of investment in the energy sector of developing countries is also a problem, they often struggle to attract the necessary investment to build and maintain their power grids. This can lead to limited infrastructure, obsolete equipment, and insufficient capacity to meet the growing demand for electricity [Inta].

3 Related Work

In this chapter we analyze and explain the overview of related work. The Section 3.1 discuss different deep learning based approaches for multiple problems in power systems. The Subsection 3.1.1 describe deep learning techniques for electricity demand forecasting in residential areas. Further, Subsection 3.1.2 explores data-driven methods for load shedding in power networks. The last Subsection 3.1.3 of Section 3.1 presents two studies that discuss fault prediction in power grids using data-driven methods.

The Section 3.2 demonstrate the history and structure of LSTM based-deep learning model and summarizes that they are beneficial for time-dependent learning applications. Further, Section 3.3 discuss hybrid approaches and indicate the benefits of combining data-driven along with knowledge-driven models. This chapter ends with the Section 3.4, which explore several studies related to AI planning, and especially it provides an outline of the advanced Temporal based AI planning research area.

3.1 Overview of Data-Driven Approaches in Power Systems

The study explores the increasing complexity of power systems data and its practical applications in [KLWK20]. It emphasizes how deep learning is becoming a more significant machine learning strategy than other approaches. The study delves deeply into both supervised and unsupervised deep learning approaches, and even makes a passing mention of semi-supervised and reinforcement learning. It emphasizes the potential of generative and discriminative deep models in particular. What is important to remember is that deep neural network optimization has enormous potential for advancing power systems research going forward.

Three main classes are identified for deep learning algorithms in power engineering [KLWK20]. In order to establish complex nonlinear decision boundaries within power system data for classification and regression tasks, Discriminative Deep ANNs fall under the first class. Probabilistic Deep ANNs, the second class, concentrate on feature learning to find hidden variables that characterize the data's probabilistic density function. The third class comprises algorithms for Deep Reinforcement Learning (DRL). These algorithms use a reward system to learn policies based on input from the environment. DRL is used in smart grids for voltage control, demand response, and lowering uncertainty in renewable generation.

Discriminative Deep Learning Applications in Power Systems, including information on the dataset, model, and outcomes, are shown in table number 1 of [KLWK20]. Applications-based grouping of different power system studies is presented in the table. We are particularly interested in fault prediction and power fluctuation identification, where two models—ReLU and LSTM—are used. Table 2 illustrates the limitations of ReLU ANN in terms of directly capturing spatial and temporal patterns as well as feature coherence, as mentioned in the section "Advantages and Restrictions of Deep Discriminative Modeling" [KLWK20]. On the other hand, the table shows how well the LSTM model performs when it comes to extracting accurate temporal features and supporting flexible input dimensions. It is designed to capture the sequential patterns in time-dependent power systems measurements.

Since in our particular application (predicting random power outages), the temporal feature extraction and the sequence/timing in which the events happen is important so we study the LSTM model intensively for fault prediction.

The literature in the subsequent subsections will extensively look into the research areas that are close to our data-driven problem of predicting unscheduled power outages.

3.1.1 Deep Learning Approaches for Residential Electricity Demand Forecasting

Son et al. addresses the challenge of forecasting monthly residential electricity demand, which is essential for power-system planning [SK20]. It introduces a forecasting model using deep learning, specifically long short-term memory (LSTM), and incorporating social along with weather-related variables. The model was validated using 22 years of data from South Korea and outperformed four benchmark models, showcasing its exceptional forecasting accuracy. This research holds potential for enhancing decision-making in power-system planning and resource utilization efficiency. Key point from the study was that the LSTM model outperformed the four benchmark forecasting models (SVR, ANN, ARIMA, and MLR).

Son et al. discusses that previously statistical methods and data mining techniques were employed to resolve this problem, and the data mining techniques have demonstrated better results at forecasting residential electricity demand because they are non-linear [SK20]. So learning from the data-set is essential as the variables that affect are unknown and could be linear or non-linear. The study in its literature review highlights various deep learning based approaches applied for electricity demand forecasting problems with different resolutions such as minute-by-minute, hourly, daily, weekly, monthly and yearly. Similarly, Son et al. proposed a method to provide a precise model for the one-month-ahead forecast of electricity demand in the residential sector [SK16]. Hence, there is a wide range of choices to select the resolution depending upon the availability of data and speed or suitability of prediction processing blocks. Hossen et al. categorizes load forecasting into three main types: long-term (1 to 10 years) for system planning, medium-term (1 month to 1 year) for efficient power system operation, and short-term (1 hour to 1 day or 1 week ahead) for estimating load flows, as outlined in the first reference of this study [HPA+17].

Son et al. showed seasonal and monthly patterns with a direct impact of temperature variables [SK20]. According to several previous studies, in the demand series is affected by several non-linear variables, such as social and weather conditions [SK20]. The introduction section cites 8 different papers and points out that consumption is influenced by many non-linear variables namely weather, economics and demographics[SK16]. In the social factors include holidays, workdays, real electricity price (REP) and consumer price index (CPI), while weather variables that affecting electricity demand consisted of wind speed, vapor pressure, mean temperature, maximum temperature, global solar radiation, cooling degree days and daylight time [SK20]. Main findings from the results of this paper were that the operation of air condition or heating appliances depends on the level of hotness or coolness of the weather. Moreover, social factors like CPI (general inflation of prices for all goods and services) and REP does not cause much fluctuation while weather does quite more in comparison. Thus, consumer is more concerned about their comfort in comparison to cost.

Another study conducted in North Macedonia in the same application area shows similar findings [VAKP20]. In this paper neural networks were used to predict short-term electricity consumption and power system load, with air temperature, day type, and humidity being crucial factors. Accurate load forecasting for the day ahead allows the system operator to plan energy supply effectively. The neural network was trained using average daily loads and temperatures as historical data, enabling it to forecast load accurately based on the relationships it learned during training. The case study input data included hourly power system load. Veljanovski et al. emphasized that climate and calendar parameters such as temperature, day type and humidity, wind speed have highly significant impact on electricity consumption and power system load on short term [VAKP20]. Because of the year seasons in Macedonia high variations of consumption and load is observed.

Raza et al. presents a new model using a feed-forward artificial neural network (ANN) with a global best particle swarm optimization (GPSO) algorithm for training [RNHB17]. The model uses meteorological data, exogenous variables, and lagged load data as inputs. Further, it adds that accurate load forecasting is vital for smart buildings and power systems, as good load forecasting leads to uninterruptable power supply for consumers. It describes that Kalman filter-based, statistical and regression models provide accurate forecast results under less uncertain load demand. Nevertheless, as the exogenous variables (human impact, cultural and sociological events) and meteorological variables (temperature, relative humidity, dew point and wind speed) are added then the errors in the above models increase. Hence, we need AI-based techniques for this kind of nonlinear input models, because AI techniques have the ability to solve complex prediction problems with high accuracy under uncertain conditions. Also, Raza et al. emphasize in its “weather input variable” section that impact of meteorological conditions on load demand has been reported in many studies [RNHB17]. It further adds that power demand rises with increase in dry bulb or air temperature.

This section’s literature review emphasizes the utility of deep learning models, the significance of social and meteorological variables, and the requirement for AI methods to manage complex, non-linear input factors. The studies also highlight the significant influence of temperature and meteorological factors on electricity consumption. Furthermore, the supply and demand of electricity are directly correlated, which causes imbalances in the power systems of developing countries.

3.1.2 Exploring Data-Driven Based Load Shedding Techniques in Power Systems

Li et al. proposes a hierarchical data-driven method for predicting event-based load shedding (ELS) to address voltage instability in power systems [LXR20]. It efficiently classifies shedding locations and predicts shedding amounts using a weighted kernel extreme learning machine. The method is highly accurate and performs well, even with imbalanced data distribution, as demonstrated on the New England 39-bus system. While citing another paper in the Introduction section it says that Load Shedding control is an effective countermeasure against voltage instability. This is why this study predicts shedding amounts and also classifies shedding locations to stabilize voltage and thus ultimately balancing the power grid.

Astal et al. says that many developing countries, including Gaza Strip, suffer with chronic electricity shortages that hinder economic development [AAAF22]. To manage this, the local utility employs daily power load shedding, but variations in demand and sources pose challenges. This article introduces a highly accurate load demand prediction model based on an ANN trained with supervisory control and data acquisition (SCADA) data. The model forecasts energy consumption and determines the daily power load shedding schedule, offering quick and precise decision-making support. Evaluation confirms its good estimation accuracy.

In the feature identification section, based on literature review it was discovered that severe weather disturbances have a disruptive impact on the entire electricity supply chain, encompassing generation, transmission, and distribution [AAAF22]. Further, a drop in temperature and humidity leads to drop in electricity demand, relevant for summers only. Moreover, the efficiency of electricity production plant also decreases with increase in temperature, this increases the probability of power outages.

The relief algorithm [KR92] is used to select and limit features and the table number 2 in the paper ranks the features in comparison to effect on electricity [AAAF22]. By analyzing the table it is clear that the prominent factor that affects the overall electricity demand is temperature.

To sum up, these studies highlight the significance of load shedding as an approach to deal with persistent electricity shortages and voltage instability. They also highlight how weather, particularly temperature, has a big influence on the supply and demand for electricity. It is highlighted that the use of data-driven models, is a useful tool for precise load demand prediction and decision support.

3.1.3 Fault Prediction with Data-Driven Techniques in Power Systems

Zhang et al. proposes a method for predicting line trip faults in power systems using LSTM networks and SVM [ZWLB17] . LSTM captures temporal features from multi-sourced data, while SVM handles classification. To combat over fitting, dropout and batch normalization layers are included. Experiments using real data confirm the method's improved performance compared to existing techniques, making it suitable for practical applications. Further, in the problem statement it states that line trip faults are common and leads to massive blackouts.

Furthermore, it adds that fault prediction involves analyzing and mining historical data to forecast potential issues in the power system, allowing for preventive measures and ensuring system safety and recovery. This technology is more advanced than fault diagnosis as it aids in making informed decisions and preventing faults. By citing two previous studies it defines that the line trip faults are frequent occurrences in power systems and have received significant research attention in recent years. When reclosing attempts fail, they can lead to extensive power outages and property damage. In the literature review it was mentioned that various AI methodologies were proposed comprising expert systems, Bayesian networks, rough sets, petri nets and neural networks. Several other studies were performed which indicated how line trip identification research progressed, for instance, improving fault diagnosis accuracy based on enhanced temporal constraint network by treating it as an optimization problem. However, these methods are limited in addressing power system faults and can be negatively affected by relay protection and electrical component malfunctions. The key point is that these methods approach the problem after the fault has happened, it unfortunately cannot do much in predicating whether there will be a fault in the power system. And we want to predict the faults in advance just like the problem in [ZWLB17].

Electrical measurement data serves as initial information about potential faults [ZWLB17]. Analyzing historical data enables the prediction of power system faults, and in our case the loss of power supply. This is why we used the historical data of real-time voltage to make predictions for the future. Recent studies have used artificial neural networks (ANNs) for fault prediction. These include an improved method using multilevel genetic algorithms for more accurate fault forecasting, an ANN-based approach for early online prediction of transient instability, and a new ANN-based methodology to detect and predict faults in a power plant's boiler burner system. Though, there is great quantity of temporal information involved during the power transmission and distribution, which contributes a significantly to fault prediction but cannot be mined by ANNs. Hence, RNN was used since it has shown strong ability to capture hidden correlations, for instance in image captioning, voice conversion and natural language processing, these paper were cited in [ZWLB17]. They also have shown good performance in dealing with faults [MS08; XWL+16]. In the results section Zhang et al. proves the link with temporal information between the line trip faults and measurement data can be mined for fault prediction [ZWLB17].

Wen et al. addresses the challenges posed by fast and stochastic power fluctuations due to renewable energy sources and flexible loads, which impact the frequency performance of modern power systems [WWT+19]. It presents a deep learning approach based on LSTM RNNs to identify real-time active power fluctuations. This method offers more accurate and faster estimation of power fluctuations, facilitating improved frequency control. The study uses a Singapore power system model integrated with distributed energy storage systems to validate the method, demonstrating its necessity and advantages over classical approaches. While referring to particular citations it says in the introduction that LSTM algorithm is highly valued for its accuracy and exceptional ability to abstract features from input and output patterns, whether these patterns are linear or nonlinear.

Further, they input the real-time measured frequency signal as input to the model, and then the identification tool provides a control reference to regulate power system frequency [WWT+19]. Technically, the goal is to regulate the frequency at nominal value (50 Hertz). Afterwards, the frequency is maintained within a stable state by synchronous generators and energy storage systems (ESSs) for the whole Singapore power system. Similarly, in our work we have maintained the power system of a smart home by using ESSs inside the residential sector. This makes the smart home unaffected by problems of the unreliable power grid. Further, it investigates that the present frequency fluctuation may depend upon previous power variations. Likewise in our case, the present outage might indicate supply / demand issue for a specific time of week / day or on specific weather etc.

While explaining the Singaporean power grid explains the proximity of the country to the equator. This leads to a greater reliance on air conditioners that impact total load demand [WWT+19]. In the same way, we collected the data-set from Pakistan that is a hot weather country and power fluctuations are common in summers compared to winters because of cooling appliances.

In conclusion, these studies demonstrate that LSTM and other deep learning techniques are well suited for anticipating and managing faults in power systems. The dependability and stability of power systems depend heavily on the capacity to foresee and prevent problems before they arise. Understanding and resolving power system issues also greatly depends on geographic and environmental variables. Hence, analyzing historical data enables the prediction of power system faults, and in our case the loss of power supply. This is why we are using the historical data of real-time voltage and frequency along with meteorological data for a specific location to make predictions for the next day.

3.2 Temporal based-LSTM Networks for Random Power Outage Prediction

First we explain the difference between ANN, RNN and LSTM in simple terms, and then we discuss the benefits of RNN and LSTM. Also, when the application of LSTM becomes necessary and is useful.

ANN is a multilayer feed-forward neural network that employs backward error propagation [MDG+19]. The Figure 3.1(a) shows a typical structure as an example. The network is made up of an input layer, hidden layers, and an output layer, as can be seen.

Traditional ANN models are not useful when it comes to modeling time-series data because they are unable to link information from one moment to the next [MDG+19]. As seen in Figure 3.1(b), in order to influence the weights at the following moment, the output of the previous moment serves as the input for the subsequent moment. This increases the accuracy of the prediction while also exhibiting temporal dynamic behavior for a time sequence.

The long-term dependencies in the input sequences are difficult for RNN to capture. Furthermore, it could lead to issues with exploding and vanishing gradients. To address this issue, Hochreiter et al. introduced Long Short Term networks [HS97]. The unique feature of LSTM is that it incorporates self-connected units that enable a value (forward pass) or gradient (backward pass) that flows into the unit to be preserved and subsequently retrieved at the required time step [MDG+19]. The specific connections between neurons are represented by the green arrows in Figure 3.1(c). They assist in preserving information in memory for a long time and are used to control the cell states [MDG+19].

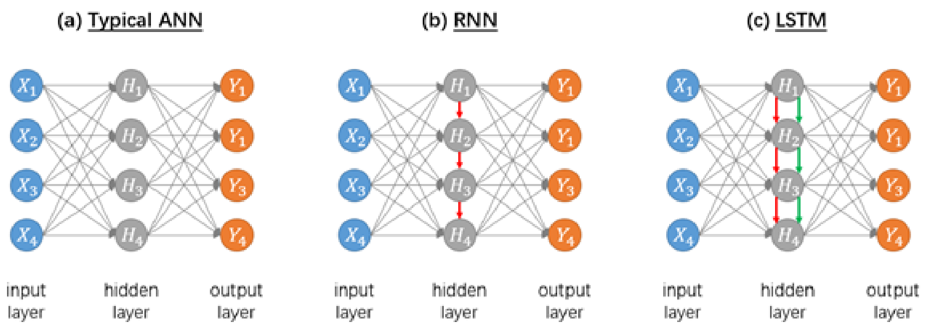


Figure 3.1: Structure of ANN, RNN and LSTM. Under CC license from [MDG+19]

ANN is the basic data-driven learning method, however, in the power transmission and distribution or electrical power grids enormous amount of temporal information is involved that can contribute greatly to fault prediction [ZWLB17]. But it cannot be mined by ANN techniques as explained above. On the other hand, RNNs, a deep learning method, have shown to extract and capture the hidden connections of features (input data) with temporal points [WWT+19; ZWLB17]. As proven in various big data applications cited by [ZWLB17], such as; image capturing, natural language processing, voice conversion and additionally, they have performed well when dealing with faults. Though, the original RNN has the problem of vanishing gradient as the later nodes perception of the previous nodes decreases, also briefly explained above. The LSTM network is generally a better RNN because it can handle longer time series [ZWLB17], fundamentally it is an RNN but an improved version. Furthermore, Zhang et al. explains that when dealing with complex operations, hybrid faults, and significant noise, LSTM networks perform well in terms of diagnosis and prediction. In a different study cited by the paper, the use of an LSTM network was proposed as a means of achieving timely fault detection and identification using readily accessible measurement signals [ZWLB17]. The findings indicated that, compared to the convolutional ANNs, the LSTM network was more effective at identifying and detecting faults in railway track circuits. In a study referenced, a deep recurrent formulation of a LSTM network is presented as a supervised temporal feature extractor to model the sequential behavior of the time-dependent power systems measurements [KLWK20]. However, the study of LSTM networks for data-based fault prediction in power systems is still in its early stages.

In order to address the vanishing gradient issue, a long short-term memory block is added to the RNN to store values for both long and short time periods [ZWLB17]. The RNN's hidden units are specifically replaced by LSTM blocks with three gates that control the flow of information into and out of their memory [HS97; ZWLB17]. A single LSTM block consists of a cell and three gates: input, output, and forget gates. The timing of allowing errors into or out of the block can be learned by LSTM networks. No values are allowed to enter the block when the input gate's weights take a value of zero. Additionally, the value is not permitted to exit when the output gate accepts a value of zero. The value is trapped in the cell when both gates are closed, meaning that it cannot increase or decrease or affect the output of the current time steps. The gradient can therefore be propagated back across many time steps during the back propagation process without blowing up and disappearing. Long short-term memory (LSTM) networks perform better in practice compared to the original RNNs because they are better able to learn the long-range dependencies of time series.

In summary, the studies mentioned suggest that ANNs are not suitable for capturing temporal dependencies in time-series data. Due to their superior sequence handling capabilities, RNNs—particularly LSTM networks—have shown to be useful in a variety of applications, such as power system fault prediction. Considering these points, LSTM network is selected as the deep learning algorithm for predicting random unscheduled power outages in this thesis.

3.3 Hybrid Approaches

Both data-driven based learning models and knowledge-driven models have seen a great deal of research for a variety of problem solutions, however, each has advantages and disadvantages. Also, a slight increase in the fusion of data-driven and knowledge-driven approaches or we can say the hybrid approaches can be seen in more recent literature. Therefore, we have applied the hybrid approach for our particular application of providing uninterrupted power supply to the smart home in unreliable power grids along with reducing electricity cost by applying DSM-based TOU technique.

Data-driven methodologies use massive datasets to learn models for a variety of issues using data mining and machine learning methods [AALC15]. Conversely, knowledge-driven approaches make use of extensive prior knowledge in the target area to construct models using knowledge engineering and management technologies. The models in knowledge-driven systems are typically static, which is widely acknowledged as a drawback. Once defined, they cannot automatically be modified to fit changing requirements or circumstances in the real world. In real-world applications, the knowledge-driven approach is insufficient to produce comprehensive modeling for every scenario the system will encounter. This is a very restrictive limitation because it is typically not feasible to define complete modeling for every scenario or series of developments for a system. One more disadvantage is that it is weak in handling with uncertainty. Reasoning and inference are typically embedded in facts rather than speculative real-time information. While some methods employ fuzzy logics and/or probabilistic reasoning, they have not yet been fully incorporated with modeling techniques. Knowledge-driven modeling is founded on real-world observations that the list of tools and capabilities needed to complete a task is consistently quite similar. For instance, you will need a liquid container, coffee, and sugar to make coffee. There are some fundamental ideas that apply to every action for making coffee, regardless of the type of coffee used, whether milk is added, or whether brown sugar is preferred to white sugar.

In contrast, data-driven approaches can effectively address aspects such as model evolution and adaptability. Though, data-driven models suffer from some issues as well. The modeling of activities requires data for the so called cold-start problem [AALC15]. Data-driven techniques cannot function right away after deployment due to the cold-start problem that results from dependence on data. There needs to be data gathering and training of the activity model. Therefore, time is required. Transfer learning techniques do however help to lessen this issue, though they do not completely solve it. Second issue is the lack of reusability. The learned models are typically not transferable to other applications because they are directly learned from concrete feature data. The primary issue is that application-specific models, not generic models, can be created using data-driven approaches. Thirdly, the annotation problem, annotated data bases may be required (supervised learning approaches).

Azkune et al. introduces a scaleable activity recognition system that combines unsupervised learning and knowledge-based models [AA18]. It extracts frequent action sequences and uses knowledge-based models to infer activities. The system's performance is comparable to supervised learning techniques, making it suitable for real-world deployments with diverse sensor setups. Combining a knowledge-based model with a machine learning technique, Pathak et al. proposed a general method that builds a hybrid forecasting scheme by leveraging the advantages of these two approaches [PWF+18]. Claiming that there are many possible uses for this kind of approach, for instance; enhancing forecasting of the weather. Liu et al. combines symbolic planning and Hierarchical Reinforcement Learning (HRL) to generate robot policies that take cost information from Maps of Dynamics (MoDs) into account, thereby addressing the challenge of long-term robot operation in dynamic environments [LPGA23]. They HRL update the problem instance to update cost terms, which is then taken in by the domain instance to create plans.

Taking everything into account, in knowledge-driven modeling, essential things and actions remain consistent for various activities, enabling the creation of basic models based on rich prior knowledge and in less time compared to data-driven models where data is to be collected, preprocessed, learned and deployed. Similarly, data-driven approaches are really good for providing solutions in uncertain or random conditions. And when knowledge-driven is combined with data-driven approach the amount of training data is significantly reduced, because the basic cases are already covered by knowledge-driven techniques. Hence, our model is neither entirely generalized nor completely application specific.

Therefore, we propose a strategy that combines a data-driven model (ML model) with a knowledge-based model (AI Planning) to create a hybrid model. Which get the benefits from both the approaches, and helps in solving our problem.

3.4 AI Planning

Piacentini et al. solves a novel class of metric temporal planning problems that involve simultaneously managing plan trajectory constraints and uncontrollable numeric events, posing challenges not addressed by existing planners [PAFL15]. They introduced creative planning techniques to address these issues. Voltage control in Alternating Current (AC) electrical networks is used to illustrate the approach. Additionally, they proposed a lookahead heuristic that, in deciding on useful actions, takes into account the interaction between active trajectory constraints and the upcoming uncontrollable event. They also extended the planner POPF2, to POPF-TIF, so that it can comprehend TIFs and incorporate them in its calculation for the plan generation. The heuristic function takes into account the upcoming violation of the system which is gets by the injection of TIFs in the problem instance of the AI planning. Then it relates it with the active trajectory limitations and stabilizes they system by applying various helpful actions defined in the domain instance of AI planning. Experimental results indicate that their approach scales effectively with network size and the number of controllable components.

Georgievski et al. highlights the significant energy consumption of buildings and the need to enhance their efficiency, particularly in the face of rising energy costs and environmental concerns [GSA23]. The goal of their work was to create an AI planning system that would increase the energy-cost effectiveness of non-residential smart buildings. The system uses AI temporal planning and is diligently designed in accordance with best practices for software engineering. It creates adaptable plans to maximize building operations by taking into account variables such as battery consumption, day-ahead energy market prices, building loads, and environmental conditions with the help of TILs and TIFs. The study shows that this strategy can reduce energy costs by an average of 43% when compared to a baseline cost through scenario analysis.

Further, the paper shows the action for charging the battery at a low price. The rate at which the battery charges depends on the cost of energy. The rate at which the battery charges and discharges as well as the energy consumption of the building determine the SOC's value. When energy costs are high, the battery is not charged during peak hours. If the SOC is too low, the battery only charges at nominal or low energy prices. The system takes in the day-ahead energy prices from European Network of Transmission System Operators for Electricity (ENTSO-E). The price of the electricity change throughout the day, also the building's energy consumption is variable and keeps changing because of multiple factors. So these variables are represented by Timed Initial Fluents (TIFs), as they TIFs can have the value in numerical form at any time throughout the day.

Another related study addresses the challenge of optimizing energy consumption in HVAC systems, which account for over 50% of a building's total energy use [Sha21]. It focuses on creating automated schedules and plans for HVAC components to maintain thermal comfort while minimizing energy expenditure. The research employs automated temporal planning, starting with a smaller air conditioning system and later applying it to a standard HVAC system. Experimental results indicate that the HVAC system using their scheduling model with automated planning consumes significantly less energy while providing the necessary thermal comfort, compared to a system without this optimization. Due to varying HVAC operating modes, that is the intermittent nature, large commercial HVAC systems benefit from scheduling that includes intervals, non-overlapping actions, time windows, and action duration restrictions. It is possible that some components must operate concurrently, necessitating concurrency and actions with time-initial-literals (TILs) [Sha21].

The Piacentini et al. have successfully applied TIFs to the problem of power system AC voltage control issues. As they considered the whole power system network from the perspective of network operator, hence, the potential remains for the research in smart homes. Also, the considered power system network was assumed to be reliable in [PAFL15].

Georgievski et al. have shown that their planning system dramatically lowers energy costs when compared to a baseline system, suggesting that TIFs are a successful way to address the issue of cost-effective energy management in buildings [GSA23]. Thus, the potential to investigate smart home remains there. Also, the ones that are located within a unreliable power system.

Shaikh investigated the application of AI temporal planning to HVAC energy optimization [Sha21]. With this method, HVAC components can be designed to achieve thermal comfort with the least amount of energy consumption. The possibility for further research persists because the modeling was limited to the HVAC system under normal power grid conditions.

Bajada investigated the application of AI Temporal Planning and improved the planning methods to successfully handle complex numerical behavior and continuous change [Baj16]. In the same work as a case study, it automates demand dispatch for electricity load management utilizing AI temporal planning, optimizing costs from the standpoint of an aggregator. It defines that the aggregator's job is to meet customer demand while maintaining adherence to operational constraints, like maintaining load levels within predetermined bounds and balancing supply and demand. Hence, loads are taken into account in this study, but from the standpoint of an aggregator rather than a smart home.

Aiello et al. describes the case study "Following Smart Grid Signals" as buildings interacting with grid by receiving signals from it and maximizing energy-related operations by employing them [AFG21]. According to their findings, a smart energy system that combines the two signals finds a balance between cutting expenses and reducing carbon emissions, allowing for cost savings while also enhancing the sustainability of the house. Some uncertainties, such as variations in weather, demand, electricity emission factor, renewable production, and electricity prices, are mentioned when describing the optimal scheduling problem [AFG21]. To counter these uncertainties, they schedule the calculations in small time steps, thus forecasting uncertainties by sequentially making short-term decisions and correcting the course of their actions based on fresh information. Though, in our work we are addressing some of the uncertainties such as changes in the weather and demand.

Some of the studies reviewed have provided solutions from the utility's perspective, while some implemented AI temporal planning to increase efficiency of smart buildings and offices, and one study designed the solution for a HVAC device. Therefore, our research seeks to explore and address DSM techniques using AI temporal planning in a smart home (residential unit), and most importantly which is located in a unreliable power grid.

4 System Design

In this chapter, we explain the system design of our study. This portion provides a comprehensive overview of our research solution's structure and components.

4.1 Architecture

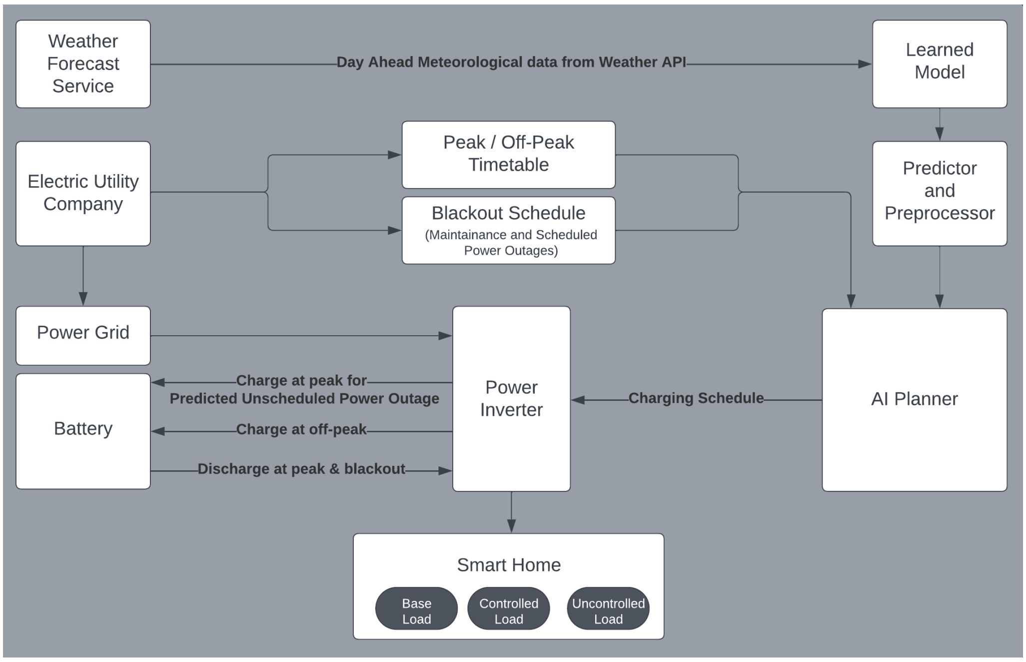


Figure 4.1: Architecture Diagram

The high-level system architecture is displayed in Figure 4.1. An AI planner, different types of loads, a data-driven Predictor block, a power inverter, a battery, and a grid connection are all included in the conceptualized smart home setup. The last subsection "load profile" provides an explanation of the load profile and its purpose, which is derived from a real-time load profile of a real house.

The weather forecast service of the city in which the smart home is located provides the exogenous weather variables to the learned LSTM based model. This meteorological data from the weather application program interface (API) along with the learned model predicts the random power outages for the upcoming day. This day ahead information of the very possible unscheduled power outages is sent to the Preprocessor block. The Preprocessor already has the roster of the scheduled power outages from the local electric utility company. After getting information of both the scheduled and unscheduled power outages, it calculates the state of charge (SOC) decrement factor for each of these power outage hours and calculates the remaining SOC. The preprocessor also gets the peak / off-peak time table from the local utility company. Then the preprocessor generates the problem file with all necessary information, and ultimately the AI planner creates a plan by taking the newly generated problem file and established domain file as inputs. As a result, the AI planner gives the charging schedule to the inverter while considering the TOU prices along with timing and tries to charge cheaply as much as it can. AI planner signals the inverter when to charge and when not to. This is because the inverter is a non-intelligent device. The charging schedule has constraints such as, should not charge during peak hours and cannot charge during power outage, can charge during peak if predictor predicts a unscheduled power outage during or just after peak hours.

A power inverter is an electronic device that converts direct current (DC) to alternating current (AC), though more recent inverters are also capable of performing the opposite operation, called Rectification. When backup is not required, it charges the batteries; when it is, it discharges them to drive the load. The inverter actually is placed between the load and the battery because the batteries store energy in the form of DC, and the inverter must use DC to charge the batteries. While the grid supplies the AC power needed to charge them, the inverter converts the grid's AC to DC so that the batteries can be charged. Because the load of the smart home runs on AC, the inverter converts the stored charge from DC to AC to run the load on the batteries when required. This causes the batteries to discharge. Further, the power grid connection is the AC power source that the electric utility provides to the smart house. The battery setup is assumed to be 20 KW considering the load profile. The smart home has three types of loads that are explained in the last section of this chapter.

Different utility companies specify their TOU-based rates and schedules, then make them available to their customers. Likewise, IESCO defines its peak/off-peak times [Isld]. Which is discussed in detail in Section 2.1.3. Also, the energy and water services regulatory commission of the Republic of North Macedonia defines peak and off-peak tariffs separately [Ene], from 22:00 to 07:00 hours and all day on Sundays the electricity is provide at off-peak rates [Bal]. Similar to this, we have defined the peak period for our system, which is the five hours from 6 to 11 p.m. The remaining 19 hours would be at a lower rate, the so called off-peak tariff. Additionally, a predetermined schedule for power outages that are brought on by the various factors covered in Section 2.8 was established earlier. The real-time demand side management is presented in the third section of [KMSM16]. In order to have real-time control of various loads, authors in the figure and explanation incorporate the blackout schedule (in this case, a power outage schedule) into their predictive scheduling layer. In the same way, our system architecture gives the AI planner the information about scheduled power outages which it receives from the utility.

So the peak/off-peak time period and scheduled power outage schedule (caused by either maintenance or load shedding) is known already. Moreover, with the help of our LSTM based random power outage predictor, we also know when the unscheduled power outage would happen. Hence, the discharging schedule in terms of time is pre-defined because of the predictor and preprocessor block. We call these two time (scheduled and unscheduled outage) periods as significant hours. During the peak-hours, the inverter shifts the load to batteries via a simple source change function, as the peak times can be put into its setting. While during a power outage, whether scheduled or unscheduled, the inverter switches the power source to batteries as the primary power source, when the power grid is absent.

Hence, this conveys the AI planner at what times the power outage will occur and what the peak hours in a day are. Hence, the AI planner is getting all the information, from which it then makes an intelligent plan and instructs the inverter to charge the batteries, with the goal of electricity cost saving and providing completely uninterrupted power supply.

4.2 Dataset Preparation

Data preprocessing is the first stage of data preparation, where raw data is cleaned to remove errors and inconsistencies, normalized to put data into a consistent format. Also includes altering it structurally to reorganize data, so it can be merged with some other data to create a constant and identical format of the dataset that is used as input to a training model.

Raza et al. recommends that many issues can be resolved by training the model using highly correlated and preprocessed data [RNHB17]. It adds that input data with outliers can harm load forecasting model performance, causing computational inefficiencies and prolonged network convergence. Furthermore, Zhang et al. depicts in the comparison of experiments that multi-sourced input model clearly improves the accuracy in relation to single input model [ZWLB17]. It shows this by drawing accuracy curves in the figure shown in the paper.

Son et al. introduced a method to identify key variables for forecasting residential electricity demand by comprehensively considering factors that can impact the demand series [SK16]. They used certain variables such as temperature, humidity, wind speed, dew factor etc. and some social factors as well. This was done to forecast electricity demand for residential sector using weather and social variables. It is important for us since increase in electricity demand leads to decrease in supply and this ultimately leads to blackouts. Thus, we use almost the same weather variables to predict random loss of power supply in the electricity network.

By considering the results and suggestions of the relevant studies, we preprocess our data and also use multiple inputs (features) to train our deep learning model. Though, most importantly we are also considering the temporal information (timestamps) of the features. There are two parts of the dataset that we have collected, one is the power grid dataset which is collected from a house based in Islamabad. Second part is the historical weather data of Islamabad. The subsequent subsections explain this in detail.

4.2.1 Power Grid Dataset

We have collected the power grid data from a house which is at the edge of the power grid network. The dataset has data from starting from April, 2022 to July, 2023. This means we have data for 16 months that is taken after every 5 minutes. Thus, each hour has 12 data entries, and a single day consists of 288 data entries. As seen in Figure 4.2, we have the timestamp in the first column (year-month-day-hour-minute-second format), then the power grid voltage in (Volts) and finally the frequency of the system in (Hertz). It can be seen in the Figure 4.2 that the timestamp starts from the end of the day, that is; 24th hour and then proceeds in the descending order. For this we run a Python based script that inverters the rows of each excel file, and sets them in ascending order. Since the data could only be imported from the server for each day, so afterwards we merged the daily data and created a single excel file for each month, using a merger script in Python.

1	Timestamp	Grid Voltage(V)	Grid Frequency(Hz)
2	2023-04-12 23:59:08	240.0	50.1
3	2023-04-12 23:53:25	239.0	49.8
4	2023-04-12 23:47:37	241.2	50.5
5	2023-04-12 23:41:53	240.1	50.2
6	2023-04-12 23:36:04	240.7	50.3
7	2023-04-12 23:30:24	239.6	50.0
8	2023-04-12 23:24:19	239.5	50.1
9	2023-04-12 23:18:43	238.6	49.8
10	2023-04-12 23:12:45	238.1	49.9

Figure 4.2: Excerpt of the historical Power Grid Dataset

Further, the data is recorded every 5 minutes, but it is not systematic, sometimes it misses to upload power grid data to the server, sometimes they were server faults or internet disconnection etc. So the entries are like, one at 6th minute of an hour, then at maybe 12th, then 17th. And on the next day in the same hour they could be like, 1st entry at 0th minute of the hour, then 5th minute, then 10th and maybe then 17th. For instance in the Figure 4.3 the timestamps are 5 minutes apart till the 170th entry at time 08:18:47, then it has a timestamp at 08:31:17, almost after 13 minutes.

166	2023-04-12 08:55:21	237.4	49.8
167	2023-04-12 08:43:14	238.7	50.1
168	2023-04-12 08:37:08	240.1	50.2
169	2023-04-12 08:31:17	238.1	49.5
170	2023-04-12 08:18:47	240.1	49.9
171	2023-04-12 08:13:36	239.9	49.8
172	2023-04-12 08:08:21	242.2	50.4
173	2023-04-12 08:02:57	242.3	50.4
174	2023-04-12 07:57:48	243.0	50.5
175	2023-04-12 07:52:21	240.5	49.7

Figure 4.3: Irregular Timestamps in the Grid Dataset

So the entries are irregular and the data has to be put in a consistent format because the weather data on the other hand is taken from a reputed website and it is in constant format. To solve this issue we ran a Python script on the monthly merged file to round off the timestamp to the nearest 5 minute interval. If the grouping needed a decision to make between two entries, then it was decided by the aggregation function to merge by the minimum value of the voltage and frequency. This is because we are more concerned with the loss of power supply, and we do not want to lose that data or round it off. For example; if we had simply rounded off the two entries and one was 240 Volts (grid ON) while the other was 0 Volts (grid OFF), the resulting merged entry would be having 120 Volts, which is false data. Thus, we use a Python based script that makes the timestamps (along with the data of that particular timestamp) regular and orderly.

Then each month file was merged into one large single excel file. Afterwards, we encoded the data based on the voltage and frequency. If the voltage from the power grid is lower than 200V, we encode that specific entry as “0”, meaning that a power outage is ongoing during that timestamp, see Table 4.1 for the rule. Since the electrical devices cannot function properly at lower voltages if the system is of 230V (in Pakistan). The power inverter also switches to battery if voltage is not in the specified range. Also, the utility has to provide it near 230V under the consumer-utility contract so anything less means blackout. So we choose the threshold at 200V, anything less means 0 (grid OFF), while value more than 200V means 1 (grid ON). This was also doubled checked with the frequency data, as the frequency also becomes 0 when there is a blackout. The Figure 4.4 shows the

Grid Voltage	Encoding	Comments
<200V	0	Power outage (grid OFF)
>200V	1	Power present (grid ON)

Table 4.1: Power Grid Data Encoding

encoding results on the dataset, column "G" is Grid Voltage, column "H" is Grid Frequency, and column "J" is the encoding result. It can be clearly seen that when grid voltage goes to 0, that is when a power outage is ongoing, the grid frequency also goes to 0 and because of the applied rule the result is 0. But when voltage is above 200V and frequency is above 45 then we consider it a 1 (grid ON).

G	H	I	J
241.5	49.7		1
241.3	49.8		1
239.5	49.6		1
238.6	49.7		1
0	0		0
0	0		0
234.5	50		1
232	49.7		1
232.1	49.7		1
232.2	49.6		1
232.8	49.8		1

Figure 4.4: Encoding of the Grid Availability

Astal et al. divided and encoded work-day and holiday with 0, 1 encoding [AAAF22]. Likewise, in the logistic regression classifier section Zhang et al. classifies and encodes the data into 0, 1 classification. Similarly, we have categorized or encoded our prediction feature into a 0/1 classification problem, because our aim is to let the LSTM model learn a 0/1 classification from the training data features.

4.2.2 Weather Dataset

The reliability and accuracy of data-driven models in power systems operation and analysis are highly dependent on the choice of data representation, specifically the features extracted from the underlying data [KLWK20]. Also, there are no specific established rules to select inputs for forecasting models [RNHB17].

We have already discussed in Chapters 2 and 3 that weather variables have a direct impact on electricity demand and faults in the power system network. We arrived at this conclusion after reviewing the findings in [AAAF22; KR92; NER; RNHB17; SK16; SK20; VAKP20; WWT+19; ZWLB17]. Temperature alone is a significant factor in making the power grid unreliable in developing countries, after temperature is humidity, then wind speed [AAAF22]. Son et al. describes that seasons in a year and temperature plays a huge role in electricity demand [SK20]. Wen et al. explained that since the country on which they performed the study lies on equator and is usually hot, so the temperature plays a prominent part in total demand [WWT+19]. Most common faults in a power system are line trip faults, which lead to un-announced load shedding or random loss of power, it can often trigger huge scale blackouts in the electricity grids [ZWLB17]. It further adds that the most frequent causes of line trip faults are weather variations, poor insulation, aging and damage to the distribution equipment.

Hence, we select have selected the feature variables to be the temperature (°C), humidity (%), wind speed (m/s), weather observed (encoded) and dew point temperature of the city from which we got the power grid data. We took the weather data of 16 months (that is; April, 2022 to July, 2023) from [Ras]. The Figure 4.5 shows the sample of weather dataset below.

Weather station New Islamabad / Liaquat Ali Khan (airport), Pakistan, METAR=OPIS,selection from 01.04.2023 till 30.04.2023, all
Encoding: UTF-8
The data is provided by the website "Reliable Prognosis", rp5.ru
If you use the data, please indicate the name of the website.
For meteorological parameters see the address http://rp5.ru/metar.php?metar=OPIS&lang=en
#

Local time in New Islaı	T	P0	P	U	DD	Ff	ff10	WW	W'W'	c	VV	Td
30.04.2023 23:30	22.0	709.1	754.6	57	Wind blow	5				No Signific	6	13.0
30.04.2023 23:00	23.0	709.3	754.6	53	Wind blow	3				No Signific	6	13.0
30.04.2023 22:30	23.0	709.3	754.6	50	Wind blow	3				Few cloud:	6	12.0
30.04.2023 22:00	23.0	709.3	754.6	47	Wind blow	4				Few cloud:	6	11.0
30.04.2023 21:30	23.0	709.3	754.6	50	Wind blow	4				Few cloud:	6	12.0
30.04.2023 21:00	23.0	709.3	754.6	47	Wind blow	5				Few cloud:	6	11.0
30.04.2023 20:30	23.0	709.3	754.6	53	Wind blow	5		In the vicinity thunder		Few cloud:	6	13.0
30.04.2023 20:00	24.0	709.4	754.6	47	Wind blow	11		Thunderstorm		Few cloud:	6	12.0
30.04.2023 19:30	26.0	709.1	753.9	45	Wind blow	5		Thunderstorm		Few cloud:	6	13.0
30.04.2023 19:00	28.0	709.3	753.9	40	Wind blow	3		In the vicinity thunder		Few cloud:	6	13.0

Figure 4.5: Historical Weather Dataset

Since the data from weather website [Ras] also includes weather observed, which is a textual representation of the weather so we coded the text into numbers, Table 4.2 shows the coding scheme.

Weather observed (In text form)	Weather observed (Encoded)
“ ” (Blank means clear weather)	0
Haze	1
Smoke	2
Mist	3
Fog	4
In the vicinity thunderstorm	5
In the vicinity thunderstorm, mist	6
In the vicinity thunderstorm, fog	7
Thunderstorm	8
Drizzle	9
Light rain	10
In the vicinity thunderstorm, rain	11
Light thunderstorm, rain	12
Rain	13
Thunderstorm, Rain	14
Heavy thunderstorm, rain	15

Table 4.2: Encoding of weather observed variable

The downloaded excel file for weather data was for divided month wise, and the entries of the data had resolution of 30 minutes for all days of the month. First, we inverted rows as the weather data was descending order, that was done using a Python script. Further, since the resolution of the power grid data is 5 minutes so we have to re-sample and extrapolate the data from 30 minute resolution to 5 minute. A Python script was utilized to re-sample the data, it copied the data six times from the previous 30 minute entry. Subsequently, all the other columns for each timestamp were updated accordingly.

4.2.3 Dataset Merging

Son et al. states that numerous variables were reviewed in the literature, but they were initially screened based on collectability throughout the entire period [SK16]. Variables were then chosen from each category based on their representativeness and how frequently they appeared in the literature. Similarly, we selected our data / features through the whole period of 16 months with a resolution of 5 minutes. This gives the LSTM model more data for each day, and this is beneficial for learning of the model. The resolution of weather data was 30 minutes, but it was resampled at 5 minutes to match the power grid data. Then both the excel files were merged to create one large dataset file with 113,201 entries for the whole 16 months. In some of the days the power grid data was not present due to various reasons, and that rows were dropped out before inputting the dataset to the LSTM model. The merged dataset with the power system dataset encoding and the extra weather variables removed is shown in Figure 4.6.

Timestamp	Temperature	Humidity	WindSpeed	WeatherObserved	DewpointTemp	Grid
2022-04-01 00:00:00	20	40	4	0	6	1
2022-04-01 00:05:00	20	40	4	0	6	1
2022-04-01 00:10:00	20	40	4	0	6	1
2022-04-01 00:20:00	20	40	4	0	6	1
2022-04-01 00:25:00	20	40	4	0	6	1
2022-04-01 00:30:00	19	43	2	0	6	1
2022-04-01 00:35:00	19	43	2	0	6	1
2022-04-01 00:40:00	19	43	2	0	6	1
2022-04-01 00:45:00	19	43	2	0	6	1

Figure 4.6: Merged Learning Dataset

The inputs to the LSTM based learning model for predicting random power outages is shown on the left side of Figure 4.7. It contains all the historical data which includes the power grid dataset along with the exogenous weather variables. All this is fed to the LSTM learning model and a trained model is created.

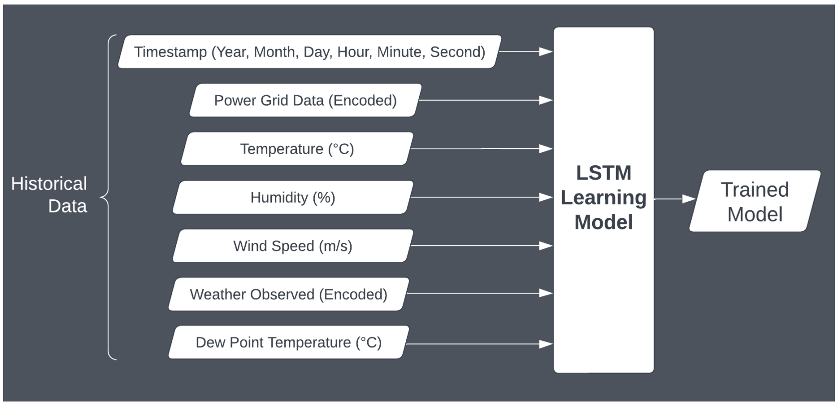


Figure 4.7: Inputs of the LSTM learning model

4.3 LSTM based Learning Model

Traditional methods like protective relays and circuit breakers react to faults after they happen, offering no prediction capability [ZWLB17]. Related to our problem, uninterrupted power supplies (UPS) or auto-start generator solutions are reactive, they start working once the power outage happens. As they are unintelligent devices and do not know at what times in the future (day ahead) the blackouts will happen so they do not preventively store energy or oil in advance. These devices are reactive in nature and we want something proactive, which can react before the loss of power occurs. We want to predict the faults or power outages in advance.

This is why in this thesis, we propose the use Long Short-Term Memory (LSTM) based Deep Learning (DL) model to predict unscheduled power outages and provide power control in the residential unit. Hence, LSTM based learned model enables proactive power outage prediction and prevention from it, enhancing the reliability and stability of the smart home.

The structure and phases of our proposed LSTM based learning model is displayed in the Figure 4.8. As explained in the Subection 4.2.1, we get the real power grid dataset from a house situated in Pakistan, likewise we get the weather dataset for the same location. Both these 16-month datasets with a resolution of 5 minutes between each row of data are preprocessed as described in-detail above, then merged together to form the learning database. Finally, the combined database is inserted into the LSTM DL mode and a learned LSTM-based prediction model is created.

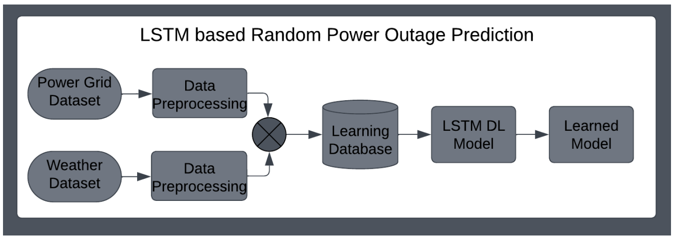


Figure 4.8: Block Diagram of the Learning Model

Since, the number of encoded 1's and 0's of the power grid voltage in our learning dataset are not balanced, the prediction results were almost always giving 1's. So the imbalance of the dataset had to be corrected, for this we used an under sampler from the Imbalanced-learn [LNA17]. This library provides tools when dealing with classification with imbalanced classes, we used the "RandomUnderSampler" method of this library to pick random entries containing 1's and 0s. As the number of rows having 0's in the "Grid" column of the merged learning dataset are 4183 out of total 113,201 entries, so the Imblearn tool randomly picks 4183 rows having "Grid" as 0 and 4183 rows having Grid as 1. Then it gives the balanced data to the LSTM model for learning. We call this model the "LSTM based Learned Model with Balanced Data".

The main contributions of our application specific LSTM model are:

- Data entries for each encoded voltage (grid present or absent), and temperature, humidity, wind speed, weather observed along with dew point temperature are chosen as the inputs to obtain more comprehensive information. The temporal information is fused with the merging layer through LSTM sub-networks or layers.

- The correlation with temporal information between random power outages (power grid data) and exogenous (weather data) is mined for unscheduled power outage prediction through LSTM networks.
- The correlation between historical random power outages (power grid data) and historical exogenous (weather data) is mined to predict day ahead unscheduled power outages using weather forecast for the upcoming day in advance.

4.3.1 Evaluation of Learned Model

In machine learning, a single full run through the entire training dataset during model training is referred to as an Epoch. In an epoch, the model updates its internal representations, modifies its parameters, and processes all of the training data. While Accuracy is a metric used to assess a machine learning model's performance, especially in classification tasks. It is typically expressed as a percentage and measures the proportion of correctly predicted instances to all instances in the dataset. The figure 4.9 presents a curve in which accuracy is shown with respect to number of epochs trained. Initially, in the first epoch the accuracy was 56%, and in the 80th epoch it was 76%.

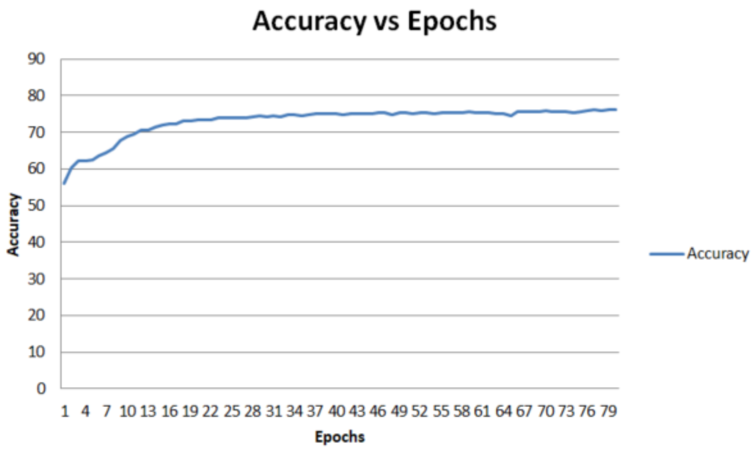


Figure 4.9: Accuracy vs Epoch graph of the Learned Model

Several classification metric terms have to be understood before analyzing the evaluating report, the terms are mentioned as under:

- Precision: Measures the accuracy of positive predictions.
- Recall (Sensitivity): Also called true positive rate, it measures the ability to find all positive instances.
- F1-Score: It is the harmonic mean of precision and recall. F1-Score balances precision and recall for model evaluation.
- Support: The number of actual instances in a class.
- Accuracy: Overall correctness of predictions. It is the percentage of predicted instances to the total instances in the dataset.

The classification report of the balanced data model implemented using "randomundersampler" method is shown in the Figure 4.10, it depicts the values of the various classification metric terms.

Classification Report:

	precision	recall	f1-score	support
0	0.73	0.82	0.77	4183
1	0.79	0.70	0.75	4183
accuracy			0.76	8366
macro avg	0.76	0.76	0.76	8366
weighted avg	0.76	0.76	0.76	8366

Figure 4.10: Classification Report for the LSTM based Learned Model with Balanced Data

Now we present the evaluation parameters of the LSTM model having unbalanced dataset (that is; all entries) for comparison purpose only, its results were not used since it was not balanced and always predicting 1's. The classification report of the unbalanced model is shown in 4.11, it can be seen that the precision, recall and f1-score for the 0s are empty or 0%. As a result, it is not able to predict 0's and it could not learn the features on which 0 could be predicted.

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	846
1	0.96	1.00	0.98	21749
accuracy			0.96	22595
macro avg	0.48	0.50	0.49	22595
weighted avg	0.93	0.96	0.94	22595

Figure 4.11: Classification Report for the LSTM based Learned Model with Unbalanced Data

In machine learning, the performance of a classification model is assessed using a table known as a confusion matrix. By contrasting the predictions with the actual ground truth values, it gives a summary of the model's predictions and help in evaluating the model's precision and its capacity to assign instances to the appropriate classes. The four main metrics that make up the confusion matrix are typically:

- True Positives (TP), the model predicted true and it is true.
- True Negatives (TN), the model predicted false and it is false.
- False Positives (FP), the model predicted true and it is false.
- False Negatives (FN), the model predicted false and it is true.

The confusion matrix of our LSTM under sampled model is shown in Figure 4.12. The TP is on the top left with 3172 predictions, TN is on the bottom right with 3164 predictions, FP is on the top right with 1011 predictions and FN is on the bottom left with 1022 predictions.

Further, on a hit and trail basis we tested our learned model with different exogenous weather variables to test the model. We found three variables on which the model is mostly dependent upon. One is the temperature, second is the dew point temperature (that is important in cases of storm and

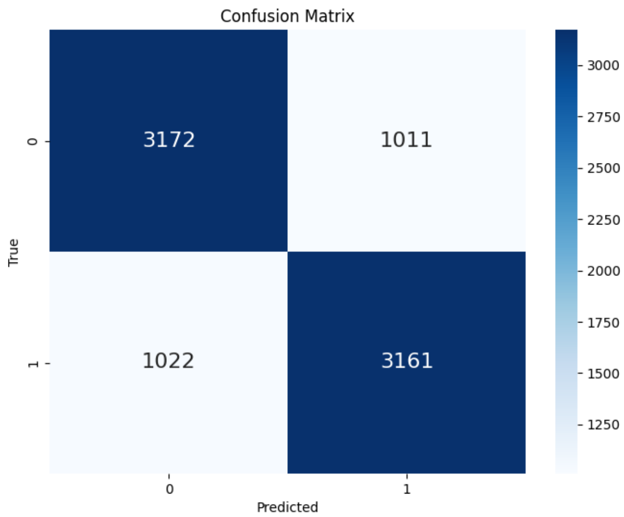


Figure 4.12: Confusion Matrix of the LSTM based Learned Model with Balanced Data

rain) and third is the temporal date/time input. The Table 4.3 show our findings when the learned model was tested with different input variables. The salient feature of this model is that the temporal information is embedded along with the features and is being learned by the LSTM model.

The array that is passed on to the predictor to predict the value of the grid is shown in the Listing 4.1, the output from the learned model can be 0 (grid OFF) or 1 (grid ON).

```
# Values read from the weather API
new_data = np.array([[2023, 6, 4, 15, 10, 0, 43, 14, 7, 3, 10]])

#For Reference ['Year', 'Month', 'Day', 'Hour', 'Minute', 'Second', 'Temperature', '
Humidity', 'WindSpeed', 'WeatherObserved','DewpointTemp']
```

Listing 4.1: Array of Input Variable for making Predictions

Input Array for Prediction	Predicted Output	Comments
[2023, 9, 24, 15, 30, 0, 30, 15, 4, 0, 7]	1	Normal weather, Moderate temp, No rain or storm or winds
[2023, 9, 24, 15, 43, 0, 30, 15, 4, 0, 7]	0	High Temperature, with normal clear weather
[2022, 8, 21, 15, 20, 0, 25, 20, 1, 0, 5]	1	Normal Weather, Moderate temperature, No rain or storm or winds
[2022, 7, 3, 11, 45, 0, 25, 80, 5, 2, 31]	0	Dew Factor Temperature and Humidity Increased
[2022, 7, 1, 5, 10, 0, 24, 100, 11, 10, 24]	0	Rainy, Windy, Stormy
[2022, 8, 21, 15, 20, 0, 25, 94, 6, 0, 24]	0	Day Time, with all other variables static, Temporal dependence
[2022, 8, 21, 21, 20, 0, 25, 94, 6, 0, 24]	1	Night Time, with all other variables static, Temporal dependence

Table 4.3: Evaluation of the Learned Model by various Inputs

Deep Learning Model	Target Variable	Precision	Recall	F1-Score	Support	Accuracy	Support	Temporal Detection
LSTM with Balanced Data	0	0.73	0.82	0.77	4183	0.76	8366	YES
	1	0.79	0.70	0.75	4183			
LSTM with Unbalanced Data	0	0.00	0.00	0.00	846	0.96	25595	YES
	1	0.96	1.00	0.98	21749			
SVM	0	0.73	0.84	0.79	837	0.77	1674	NO
	1	0.82	0.70	0.75	837			
Logistic Classification	0	0.60	0.60	0.60	837	0.60	1674	NO
	1	0.60	0.60	0.60	837			

Table 4.4: Summary of the performance results of four Machine Learning Models

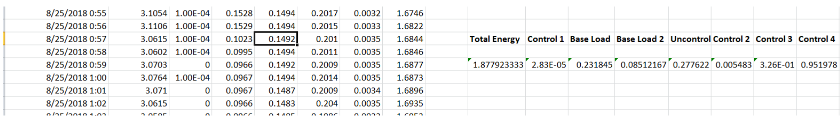
4.3.2 Comparison with other Algorithms

In this subsection the performance results of the proposed LSTM model were compared with two benchmark classification models for comparative analysis. Also, we compared our random under sampled LSTM model with raw LSTM model. The Table 4.4 depicts the various classification metric terms to evaluate the performance of the models.

The LSTM with unbalanced data model is unable to predict 0's (power outages), that is why we have selected and implemented the LSTM with balanced data model. Though, the (LSTM with unbalanced data model) has a prediction accuracy of 96% for 1's, it has 0% accuracy for 0's. The benchmark model support vector machines (SVM) also achieves a good accuracy score of 77% but it cannot capture the underlying temporal information among the features. The Logistic Classification achieves a bit poor accuracy of 60% and it is also not able to extract the significant temporal context from the features. Therefore, we have selected the LSTM model with random under sampling for this study.

4.4 Load Profile

Nadeem et al. collected electricity consumption patterns for 42 residential properties with diverse demographics [NA19]. This data was gathered over a period of one year and includes information on the overall electricity consumption of the houses, as well as data on high powered devices and major areas of the buildings. Pakistan Residential Electricity Consumption Dataset (PRECON) has hourly load profiles of multiple houses. And on the right side are the assumptions we considered for this study, which we explain in the following paragraphs.



8/25/2018 0:55	3.1054	1.00E-04	0.1528	0.1494	0.2017	0.0032	1.6746
8/25/2018 0:56	3.1106	1.00E-04	0.1529	0.1494	0.2015	0.0033	1.6822
8/25/2018 0:57	3.0615	1.00E-04	0.1023	0.1492	0.201	0.0035	1.6844
8/25/2018 0:58	3.0602	1.00E-04	0.0995	0.1494	0.2011	0.0035	1.6846
8/25/2018 0:59	3.0703	0	0.0966	0.1492	0.2009	0.0035	1.6877
8/25/2018 1:00	3.0764	1.00E-04	0.0967	0.1494	0.2014	0.0035	1.6873
8/25/2018 1:01	3.071	0	0.0967	0.1487	0.2009	0.0034	1.6896
8/25/2018 1:02	3.0615	0	0.0966	0.1483	0.204	0.0035	1.6935

Total Energy	Control 1	Base Load	Base Load 2	Uncontrol	Control 2	Control 3	Control 4
1.877923333	2.83E-05	0.231845	0.08512167	0.277622	0.005483	3.26E-01	0.951978

Figure 4.13: Hourly Load Profile

The dataset provides an excel file for each house, it has a resolution till each minute of the energy consumption. Separate columns represent different loads of the house and also the total load consumed. The exact detail of the devices in each specific house along with the number of occupants is explained in the metadata file provided along with the data set. Although the energy measuring device is attached to only specific devices, such as air conditioners in different rooms, kitchen, lounge room and basic fan/light loads. The devices such as microwave oven, electric kettle etc. are included in the column of kitchen load. For simplicity, we chose the dataset of house number 1. It has three conditioners, one in the drawing room, one in the dining room and one in the bedroom. Besides that it has columns for base load (fans, lights and power sockets etc.) in various rooms, and for kitchen and lounge rooms. Also, we took an average of 60 minutes for each column of the data (for all kinds of load) to calculate the energy consumed in KWH hour, as our model consider the load profiles on hourly basis. Since the resolution of the given data points was 1 minute, this conversion was necessary. Average computation was performed using Microsoft Excel based formulas. The load profile of 1 day was chosen for this study, it was selected by analyzing which day in that specific house has the most load consumption, so this way we could plan for the worst case and the system could then easily function for rest of the days.

Khoury et al. categorized loads into different load classifications [KMSM16]. For instance, they classified lighting, fans and TV into the “base load” category. Also, they added the HVAC system in the “scheduled load” category. And loads such as microwaves fall into the “real time controlled” load category. Similarly, we define three types of load categories, base load, controlled load and uncontrolled load category according to the features/meta-data of the specific column.

- Controlled Load: In the context of power systems it is a type of load that can be switched ON and OFF if energy management measures are to be taken. Air conditioners and heaters come into this category. In our load profile, we have the air conditioners in several rooms. So, we assume them as controllable loads. Secondly, the user will not experience inconvenience since we would be powering this load from the batteries during power outage or peak time.
- Base Load: These types of loads refer to the minimum amount of power demand required to meet basic needs of the consumers at all times. These could be fans, lights and power sockets etc. It was already mentioned in the load profile of this house, so we also consider this for our study.
- Uncontrolled Load: This type of load refers to electrical devices that operate continuously or at irregular intervals and are not designed to be switched OFF by an energy management system. The consumer just uses them whenever it is required. Examples of these loads could be refrigerators, electric kettle, TV and electric oven etc. In our case, the column mentioned with kitchen load is presumed to be uncontrollable load as it contains various kitchen devices mentioned in the previous sentence.

5 Implementation

In this chapter, we discuss the implementation of the problem.

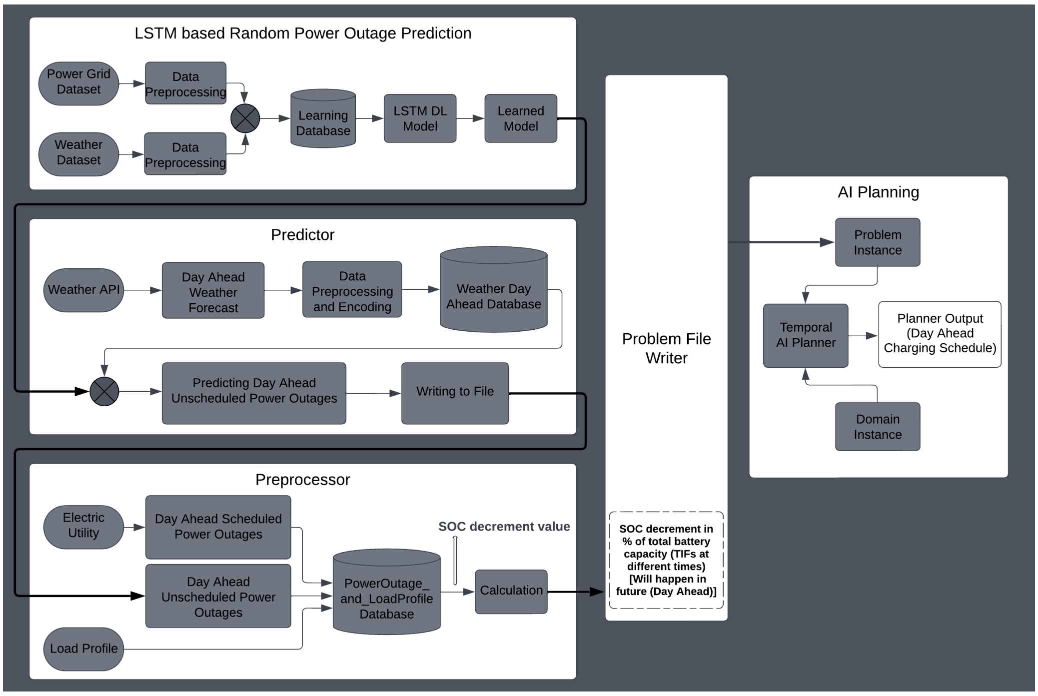


Figure 5.1: Detailed System Block Diagram

The system block diagram of our solution is shown in Figure 5.1. This section explains the approach used to solve the problem by implementing it on an AI planner. The system can be divided into four main parts. The first one is the LSTM based power outage prediction model, then the Predictor block, then the Preprocesor block and finally the AI planning part. The LSTM based model was explained earlier in the Section 4.3.

5.1 Working of the Predictor

The predictor is the part that makes the unscheduled power outage predictions. The forecasted day-ahead weather data is taken as an input in the block, it is then adjusted to our existing format, and excess variables are removed. After that, it is sent to the already learned model for making predictions. Then the predicted random outages are written in an excel file by indicating 0's (grid

OFF) in front of those hours in which unscheduled blackout will happen. While in the other normal hours 1's (grid ON) are inserted into the excel file. All these steps are performed by a Python based script.

The weather data is requested by the "get" method of a Representational State Transfer (REST) - Application Program Interface (API) in the start of the predictor's Python script. After getting the response from the weather API we extract the necessary columns since the weather API sends several weather variables. We only want the weather variables that were used as features during the training of our machine learning algorithm. Also, the predictor encodes the weather observed variable as described earlier in Table Table 4.2. Moreover, the timestamp is converted into date-time format to match the timestamp format of the training dataset. When the extraction, cleaning and conversions are done, the day-ahead forecasted weather data is written into an excel file (weather day-ahead database).

The exogenous weather variables along with their timestamps are provided as input to the trained model (saved as a Hierarchical Data Format (HDF) file), and the output from the model is Po (hour), as shown in Figure 5.2. The Po (hour) depicts that the power outage is in terms of hours of the next day, starting from the 0th hour and ending on 24th hour. Hence, the unscheduled power outages are also predicted hourly, just like the scheduled power outages. This keeps the scheduled and unscheduled power outages standardized and easily computable for the AI planning domain.

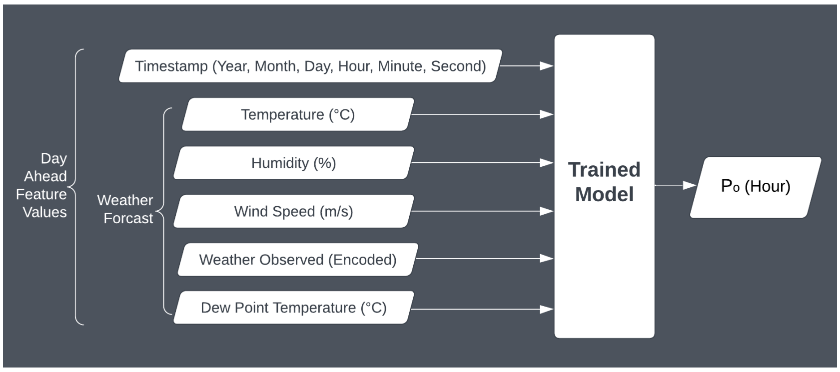


Figure 5.2: Inputs and Output of Predictor

The machine learning based prediction part of the code reads from the weather day-ahead database, and make predictions for each hour of the day ahead. Once, the predictions are performed, they are written to the power outage and load profile database as 1's (grid ON) and 0's (grid OFF) in the columns next to the hours of the day. This helps the preprocessor understand that on which hours of the day unscheduled power outages will occur, and then it makes the appropriate calculations.

5.2 Working of the Preprocessor

Preprocesssor is the part of system which generates the problem instance for the Temporal AI planning. It gets inputs from the Utility, Load profile and Predictor, these inputs when combined forms the power outage and load profile database. Remember, that this database has both the scheduled power outages provided by the local electric utility company and also the unscheduled power outages predicted by the Predictor. When this database is injected into the preprocessor, it calculates the percentage of SOC that decrease after each hour of usage, since during that hour the load would be driven on the battery. For instance, there was a power outage at 4 AM, while the battery's SOC was at 70% before that, which is the median of the range (40-100 %) defined in the domain file. The battery's SOC must always remain between 40-100% that is the mixed metric constraint. Now, from the load profile the preprocessor gets the total power consumption at that specific hour, which is 2.62 KWH. And for simplicity and according to load profile we have already pre-defined that the battery's capacity is 20 KW.

The power outage starts from 4 AM and ends at 5 AM, so at the subsequent hour that is at 5 AM the battery's SOC will have reduced somewhat as it was supplying power to the load. The preprocessor calculates this decrement value by dividing the forecasted power consumption of that hour by the total capacity of the battery, and it gets the value of the battery in percentage that is reduced afterwards. Hence, for this scenario the SOC decrement value would be 2.62 divided by 20 equals 0.1315 or 13.15%. Consequently, the battery's SOC at 5 AM would be 70 minus 13.15 equals 50.4 %. This is how the preprocessor is working. The load shifting is done by the power inverter attached between the batteries and the load as described in the Section 4.1.

Likewise, the preprocessor that is based on a Python script, can input multiple hours of power outages (scheduled and unscheduled) at different time periods, which can be set in the script earlier. Also, the peak hour time period is defined to be from 18:00 (6 pm) to 23:00 (11 pm), this was done after analyzing the peak time periods provided in [Isld], that were discussed in the Section 2.1.3, so the model is as close to the real world as possible. Similarly, the load profiles are actual real-time data of the same country, so that the external or other conditions also remain the same.

Once, the pre-processor has calculated all the SOC decrement values for different times of the day when power outage happens or during the peak hours of the day, then it re-calculates the remaining battery SOC at all 24 hours of the day, which is meaningful information for the AI planning. These numeric values of the battery's SOC computed for each hour of the day are the Timed Initial Fluents (TIFs), which are written into the problem file. The TIFs already have the load driving comprehensive information for the significant hours embedded in them that makes the TIFs quite useful to impart great amounts of information to the AI Planner POPF-TIF. Lastly, the preprocessor writes the PDDL based problem file automatically.

The PDDL domain file is fixed and it does not change, the problem file can be changed depending upon the load profile, battery capacity, scheduled power outage plan of the utility and weather conditions (unscheduled power outages). The preprocessor counts the scheduled power outages from the power outage and load profile database and then sets the value of the PDDL functions "cheap_priority_level" and "priority_value" accordingly. Likewise, the preprocessor counts the number of unscheduled power outages that were forecasted by the predictor block. Then according to the count, the preprocessor selects the suitable case in the Python based script and then updates the values of functions such as "random_run_level","random_run_capacity_value","random_run_expensive_level" and "random_run_expensive_capacity_value". We explain these functions in the subsequent Section 5.4.1.

The preprocessor also gives the TIFs as output along with certain predicates and other function values. The TIFs are represented in terms of simple battery's SOC and on the other hand the mixed metric constraint in the domain file is concerned with the battery's SOC only.

5.3 Working of the Temporal AI Planner

TIFs provide the realistic data that how much SOC of the battery will be left after each significant hour. Alternatively, we can say that TIFs would be providing imminent violations of the mixed metric constraint, because the mixed metric expression would not remain inside the bounds of the constraint as the uncontrollable fluent would be reduced. Then as explained in Subsection 2.6.2 “Enhancing Planning with TIF-based Heuristics”, the AI planner would identify and apply the helpful actions that will avoid the expected violation of the mixed metric constraint. This prevention of the constraint violation would be achieved by increasing the numerical value of the controllable fluent which is an effect of the helpful action applied. Thus, the controllable fluent summed up with the uncontrollable fluent would now be satisfying the mixed metric constraint. And the controllable fluent summed up with the uncontrollable fluent, makes the mixed metric expression, which is actually compared against a constant value as explained in subsection 2.6.2. This is how the planner makes day-ahead plans for providing uninterrupted as well as inexpensive electricity.

We define our mixed metric expression as State of Charge (SOC) of the battery. That has two parts, SOC uncontrollable part that is provided in the problem file by numeric background events happening in the residential unit. While the other is the controllable fluent of SOC, that is the adjusted value.

Both of fluents summed together make the mixed metric expression. It is possible to use mixed metric expressions to indirectly represent the relative changes of numeric variables using TIFs. For example, the controllable fluent is named “battery-soc-fix”, while the uncontrollable fluent is “battery_soc”. We want to express it in this form (at 6.0 (decrease (SOC) 10)), but we cannot express it in this syntax. So we decompose the fluent (SOC) into two fluents, an uncontrollable fluents (battery_soc) and a controllable fluent (battery-soc-fix), and replace (SOC) with the mixed metric expression (+ (battery_soc) (battery-soc-fix)). It is now possible to write the relative increase effect on (SOC) as the TIF (at 6.0 (= (battery_soc) -10)).

There is a mixed metric constraint with two constants, which define the upper and lower limit for the battery’s SOC. The goal is that the battery’s SOC must remain between the upper and lower bounds till the end of the time period, and there is no violation.

The objective of the planner is to ensure that the battery’s SOC remains within the envelope, despite the possibility of the battery depletion factor caused by running the load of the house on the battery, thus exceeding the constraint due to background events (such as peak-hours and power outages). To achieve this, the planner must select and execute appropriate actions at the right time. Those are called helpful actions in [PAFL15]. At the end, we need a plan that is automatically generated by the planner, invoking the helpful actions according to the TIFs caused by background events, thus satisfying the constraint on the mixed metric expression and solving BTMP.

5.4 Domain Instance of AI Planning Model

We describe the domain instance of the PDDL based Temporal AI Planning model in this section.

5.4.1 Predicates and Functions

The “predicates” and “actions” of the PDDL domain instance are covered in this subsection. The Listing in 5.1 displays the portion of the domain file that shows the predicates and functions created for the implementation of the Temporal AI planner domain.

```
(:predicates(complete)(begin)
(enable) (day_ended)
(peak) (off_peak)
(is_not_blackout) (charging_now)
(is_not_random_blackout)
(random_shed_during_peak)
)

(:functions
(battery_soc)
(battery-soc-fix)
(lower_limit)(upper_limit)
(charging_rate)
(cheap_priority_level)
(priority_value)
(random_run_level)(random_run_capacity_value)
(random_run_expensive_level)(random_run_expensive_capacity_value)
)
```

Listing 5.1: Predicates and Functions of the Domain File

The predicate “begin” starts the planner to plan the 24 hours schedule. It actually asserts the starting condition of the envelope action, between which all the other actions can be applied. The envelope gets closed when the “day_ended” predicate is asserted in the problem file. The purpose for the rest of the predicates are be described in the subsequent sections.

The function constructs of PDDL version 2.2 are called numeric fluents that were explained in detail in Chapter 2. They are variables which can be assigned any numeric value. The values can have both integer and fractional parts, also it can either be a positive or negative number.

The numeric fluent “battery_soc” is the uncontrollable fluent that is computed in the preprocessor and inserted to the AI planner via the problem file. While, the fluent “battery-soc-fix” is defined as a controllable fluent because it can be modified by the domain file. Thus, the “battery_soc” along with the “battery-soc-fix” makes the mixed metric expression that is shown in the following section.

```
(at 9.0 (= (battery_soc)56.852175))
(at 9.0 (not(is_not_blackout))) ;POWER OUTAGE
(at 10.0 (is_not_blackout))
(at 10.0 (= (battery_soc)50.4043333333))
```

```
(at 11.0 (= (battery_soc)50.4043333333))
(at 11.0 (not(is_not_random_blackout))); RANDOM POWER OUTAGE
(at 12.0 (is_not_random_blackout))
(at 12.0 (= (battery_soc)39.6000555))

(at 21.0 (random_shed_during_peak)) ; RANDOM POWER OUTAGE COMING AHEAD
(at 22.0 (= (battery_soc)-9.307875000000008))
(at 22.0 (not(is_not_random_blackout))) ; RANDOM POWER OUTAGE
(at 23.0 (is_not_random_blackout))
```

Listing 5.2: Predicates and Functions of the Domain File

The predicate "is_not_blackout" is created as an inverted predicate, because it was not possible to create a predicate "blackout" and then invert it using a NOT construct of PDDL in the precondition of the actions. This is because the AI planner POPF-TIF does not support negative-preconditions, as POPF-TIF is based on AI planner POPF2 which basically does not support negative-preconditions [CCFL21]. Because of this "is_not_blackout" predicate was created to cater for scheduled power outages. Likewise, the predicate "is_not_random_blackout" is created to handle for the unscheduled power outages. Though in the problem file when a power outage is to be asserted we simply use the keyword NOT as shown in Listing 5.2, which actually means there is a power outage or blackout. When the power is restored from the utility, then we assert the "is_not_blackout" predicate again. Similarly, the Listing 5.2 shows the use of "is_not_random_blackout". The predicate "random_shed_during_peak" intimates the AI planning domain that during the peak time-period an unscheduled power outage is incoming as predicted by the Predictor. This allows the appropriate action charging action to run if all other preconditions are satisfied.

Also, it can be seen from 5.2 that the assertion of the predicate is actually a Timed Initial Literal (TIL). As explained in Chapter 2, the TILs are constructs that indicate to us that at what point in time a fact becomes true which was previously false. So as the power comes and goes, we make the predicate true and false on those particular hours. The functions are explained in the succeeding sections.

5.4.2 Envelop Action

We explain the working of our envelope action in this subsection. It is the main durative action of the AI Planning domain.

The construct of a durative action is shown in 5.3. This durative action is not a normal durative action, instead it is an envelope action as explained in detail in Subsection 2.6.1. In the duration 100 is written which means that the duration of this action can be 100 if the preconditions are satisfied, but we assert it on the time unit 24, which means that the day is ended. In our modeling, we assume 1 time unit as 1 hour for simplicity. The name of the envelope action is "Day_Ahead_Plan".

```
(:durative-action Day_Ahead_Plan
:parameters()
:duration (<= ?duration 100)
:condition(and
   (at start(begin))
   (at end (day_ended))
   (over all (and
   (<= (+ (battery_soc) (battery-soc-fix)) (upper_limit))
   (>= (+ (battery_soc) (battery-soc-fix)) (lower_limit))
   ))
)

:effect(and
(at start(enable))
(at end(complete)))
)
```

Listing 5.3: Envelope Action of the AI Planning Domain Instance

The predicate "begin" is used in the precondition of this durative action, but it is used with a prefix "at start ()", this indicates that the envelope can only start when the "predicate" is true. We provide this predicate in the start, and then at time 0.1 we make it false, this can be seen in the Listing 5.4. This is done so that the envelope can open and keep running till the end of the day. Since the condition was with the construct "at start" so it only checks during the start of this durative action.

```
(define (problem CompleteUnInterruptedPowerSupplyProblem)
(:domain CompleteUnInterruptedPowerSupply)
(:init
(begin)
(at 0.1 (not(begin)))
(charging_now)
(=(lower_limit)40)
(=(upper_limit)100)
..
..
```

Listing 5.4: Initial part of Problem File

The construct "overall" in the precondition defines that during the course of this action the subsequent conditions must always be true. The uncontrolled fluent "battery_soc" is added with the controlled fluent "battery-soc-fix" to make the mixed metric expression. This mixed metric expression actually is the battery's SOC during the time period of the envelope actions (I-e: 24 hours). The comparison construct "<" is designed such that it is true if the fluent first to its right is lesser than the second fluent. The "battery_soc" and "battery-soc-fix" becomes a single fluent expression during the computation of the comparison operator inside the AI planner as both of them are inside a single set of parenthesis.

When the "day_ended" predicate which is at the end of problem file becomes true at time 24 (meaning the 24th hour), the envelope action cannot execute anymore as it is a precondition with the prefix "at end ()" so the effects of the envelope are implemented now.

The effects of applying the envelope action are two, one is that the predicate "enable" becomes true. This predicate is sort of a go-ahead signal to the other functions, that they can be applied now. This is why this effect is with the prefix "at start ()". It is a precondition for the other actions of the domain file as well, without which they cannot be applied. The other predicate with the prefix "at end ()" asserts the "complete" predicate, which helps the planner to reach the goal. As in the problem file the predicate "complete" is the condition for reaching the goal state. So, at the end of the envelope action this predicate becomes true which means that all other actions have been applied and the day has ended already because the predicate "day_ended" became true in the problem file earlier, which is why the effects are being implemented now. This is how the AI planner reaches the goal state and finally produces the day-ahead plan.

5.4.3 Actions

There are four durative actions apart from the main envelope action. Two of the durative actions namely "Battery_Charge_Cheaply" and "Battery_Charge_Expensively" deals with scheduled power outages. While the other two actions "Battery_Charge_for-Random_Cheaply" and "Battery_Charge_for-Random_Expensively" addresses unscheduled power outages.

Battery Charge Cheaply

The Listing 5.5 shows the PDDL code for the action "Battery_Charge_Cheaply".

```
(:durative-action Battery_Charge_Cheaply
:parameters()
:duration (=?duration 0.33)
:condition(and
         (at start(enable))
         (at start (charging_now))
         (over all (off_peak))
         (over all (is_not_blackout))
         (over all (is_not_random_blackout))
         (at start (>= (random_run_level) random_run_capacity_value))
         (at start (<= (cheap_priority_level) priority_value))
         (at end (<= (+ (battery_soc) (battery-soc-fix)) (upper_limit)))
         ;the line above restricts it from charging over 100%,
         )
:effect(and
(at end (increase (battery-soc-fix) charging_rate))
(at end (increase (cheap_priority_level) 1))
(at start (not(charging_now)))
(at end (charging_now))
```

```
))
```

Listing 5.5: Durative Action: Battery Charge Cheaply

This action is a durative action with a running duration of 0.33 time unit which approximately corresponds to 20 minutes, as 1 time unit was assumed to be 1 hour. The purpose of this action is to charge the battery during off-peak hours as much as it can, so fewer expensive charging actions would be applied then. Through this the cost saving could be maximized as the utility provides cheap electricity during off-peak hours. This action is particularly trigged when unscheduled power outage is to be handled. While the purpose of charging remains the same, to provide energy during the significant hours.

The preconditions for the at start construct are “enable” and “charging_now”. The “enable” predicate was explained in the previous subsection. The predicate “charging_now” is kind of a semaphore lock, it is one of the preconditions for this action, and it locks the charging resource when one charging action is ongoing. Because at a time only one action can be applied and when it is completed only then the other action can be applied or the same one can be implemented again. So in the effect of the action, at the start, “charging_now” is made false by a NOT keyword, I-e: it is locked. And at the end it is released as an “at end ()” effect of this action. Though, at the start of the problem file, this predicate is first given so the AI planner can at least apply one action in the beginning, then the locking and release mechanism can work automatically. The last condition is sort of like the mixed metric expression, this is to limit charging of the battery up till the upper limit, as charging more than 100% is not realistic in the real world.

Also, the “off_peak”, “is_not_blackout” and “is_not_random_blackout” predicates are preconditions for this action to be applied. They are pre-fixed with “overall” construct which means that for this action to run, these two conditions must hold during the whole duration of the action execution, from the start till the end. The “off_peak” predicate is used in this action because it allows charging the battery cheaply during off-peak rates only. While, the predicate “is_not_blackout” is applied here since charging is not possible if utility is not available (I-e: during a power outage). Similarly, charging is also not possible during an unscheduled power outage, this is why “is_not_random_blackout” predicate applied as a precondition with the construct “overall”. The precondition “random_run_level” and “cheap_priority_level” should be less than “random_run_capacity_value” and “priority_value” at the start for the action to execute. It tells that this action can only run if the durative action ”Battery_Charge_for-Random_Cheaply” has already ran before or the preprocessor selected a different case (no unscheduled power outage case) and the “random_run_level” was initially greater than “random_run_capacity_value”.

At the end of this durative action the controllable fluent “battery-soc-fix” is increased according to the “charging_rate” that is provided in the problem file via the preprocessor. Further, the “cheap_priority_level” is incremented each time this action is applied, it is a counter fluent which saves the no of executions.

Battery Charge Expensively

The Listing 5.6 shows the PDDL code for the action “Battery_Charge_Expensively”.

```
(:durative-action Battery_Charge_Expensively
:parameters()
:duration (=?duration 0.33)
:condition(and
         (at start(enable))
         (at start (charging_now))
         (over all (peak))
         (over all (is_not_blackout))
         (over all (is_not_random_blackout))
         (at start (>= (cheap_priority_level) priority_value))
         (at start (>= (random_run_expensive_level) random_run_expensive_capacity_value))
         (at end (<= (+ (battery_soc) (battery-soc-fix)) (upper_limit)))
)
:effect(and
(at end (increase (battery-soc-fix) charging_rate))
(at start (not(charging_now)))
(at end (charging_now))
))
```

Listing 5.6: Durative Action: Battery Charge Expensively

The duration of this durative action is the same which is approximately 20 minutes. The purpose of this action is to charge the battery during peak hours if the AI planner fears that the cheap charging actions are not enough to provide electricity during significant hours and the trajectory constraint will be violated. Basically this happens because the AI planner expects a power outage between the peak hours or just after them, I-e: during or just after the significant hours. As the peak hours are aligned in a single set of hours and are not scattered during the day, the battery level drops when the inverter is providing the energy from the battery.

In order to keep the battery SOC above the pre-defined minimum value this action is applied. So normally the system charges using cheap electricity time, but if it is informed by the TIFs that power outage is going to happen in between or just after peak hours, then, it even charges the battery during expensive electricity time and backs up the SOC of the battery. Though, because of the modeling of predicates such as “cheap_priorirty_level” and “priority_value” it attempts to charge during off-peak hours as much as it can. So the action can only run when it has already used the cheap charging action multiple times according to the “priority_value” but it still needs charge. Similarly, the modeling of predicates “random_run_expensive_level” and “random_run_expensive_capacity_value” is such that the AI planner tries to charge during off-peak hours as much as it can. So the action can only run when it has already used both the cheap charging actions (normal and Pre-Emptive) multiple times according to the “random_run_expensive_capacity_value” but it still needs a charge.

As an effect the durative action increases the controllable fluent “battery-soc-fix” according to the “charging_rate” at the end. And during the course of this durative action “peak” and “is_not_blackout” should be asserted. The predicate “peak” because this is an expensive charging action can be applied during the peak hours only, the peak hours is defined in the problem file.

Battery Charge for-Random Cheaply

The Listing 5.7 shows the PDDL code for the action “Battery_Charge_for-Random_Cheaply”.

```
(:durative-action Battery_Charge_for-Random_Cheaply
:parameters()
:duration (=?duration 0.33)
:condition(and
         (at start(enable))
         (at start (charging_now))
         (over all (off_peak))
         (over all (is_not_blackout))
         (over all (is_not_random_blackout))
         (at start (<= (random_run_level) random_run_capacity_value))
         (at end (<= (+ (battery_soc) (battery-soc-fix)) (upper_limit)))
         )
:effect(and
(at end (increase (battery-soc-fix) charging_rate))
(at end (increase (random_run_level) 1))
(at end (increase (cheap_priority_level) 1))
(at start (not(charging_now)))
(at end (charging_now))
))
```

Listing 5.7: Durative Action: Battery Charge for-Random Cheaply

The purpose of this action is to charge the battery during off-peak hours as much as it can, so fewer expensive charging actions would be applied later. This action is only trigged when scheduled power outage is to be handled. Through this the cost saving could be maximized as the utility provides cheap electricity during off-peak hours. The purpose of charging remains the same, to provide energy during the significant hours. The preconditions such as “charging_now”, “off_peak”, “is_not_blackout”, “is_not_random_blackout” were explained in the subsections above.

The precondition “random_run_level” should be less than “random_run_capacity_value” at the start. It means that this action will run firstly as the predictor forecasted an unscheduled power outage and subsequently the case selected by the preprocessor was the one which has unscheduled power outages in it. The number of power outages define the value of the functions “random_run_level” and “random_run_capacity_value”. When the level is at the capacity this action will not be able to run, meaning all the possible cheap actions handling uncertainty have been exhausted.

At the end of this durative action the “cheap_priority_level” is incremented each time this action is applied, it is a counter fluent which saves the no of executions. Likewise, since it was an action that handled random unscheduled power outage so “random_run_level” is incremented as well.

Battery Charge for-Random Expensively

The Listing 5.8 shows the PDDL code for the action “Battery_Charge_for-Random_Expensively”.

```
(:durative-action Battery_Charge_for-Random_Expensively
:parameters()
:duration (=?duration 0.33)
:condition(and
        (at start(enable))
        (at start (charging_now))
        (at start (random_shed_during_peak))
        (over all (peak))
        (over all (is_not_blackout))
        (over all (is_not_random_blackout))
        (at start (<= (random_run_expensive_level) random_run_expensive_capacity_value))
        (at start (>= (cheap_priority_level) priority_value))
        (at end (<= (+ (battery_soc) (battery-soc-fix)) (upper_limit)))
)
:effect(and
(at end (increase (battery-soc-fix) charging_rate))
(at end (increase (random_run_expensive_level) 1))
(at start (not(charging_now)))
(at end (charging_now))
))
```

Listing 5.8: Durative Action: Battery Charge for-Random Expensively

This action is meant to charge the battery during peak hours in the event that the AI planner is concerned that the trajectory constraint may be broken and cheap charging actions will not be sufficient to supply electricity during significant hours. Essentially, this occurs because the AI planner anticipates a random power outage during or shortly after the peak hours. The AI planner is intimated of this unscheduled power outage by the problem instance, which was generated by the preprocessor, and the preprocessor was fed in about this random blackout by the predictor. The battery level lowers when the inverter is using the battery to provide energy because the peak hours are concentrated into a single set of hours rather than being spread out throughout the day. This action is implemented to maintain the battery's state of charge above the predetermined minimum value.

Many of the preconditions that are same in this durative action were described in the subsections above. The precondition "random_run_expensive_level" should be less than "random_run_expensive_capacity_value" at the start. Whereas the "cheap_priorirty_level" must be greater than "priorirty_value", which forces the AI planner to firstly apply the cheap charging actions as many as it can to save cost (DSM). And if it cannot run anymore cheap charging actions, then and only then it should run this durative action. The predicate "random_shed_during_peak" is in the precondition of this action only, it acts like a flag which tells that system that an unscheduled power outage will take place during or just after peak hours, and the battery is already at low during this period because of providing energy during peak hours. So this flag permits this action to execute because it is necessary now in order to provide uninterrupted power supply to the smart home.

The numerical number of the power outages define the value of the functions "random_run_expensive_level" and "random_run_expensive_capacity_value". At the end of this durative action the "random_run_expensive_level" is incremented each time this action is applied, it is a counter variable which stores the no of executions of this specific action.

6 Experimental Design, Results and Discussion

We demonstrate the experimental design/setup in Section 6.1 and the results in Section 6.2. Afterwards, we analyze the results along with the experimental setup in Section 6.3.

6.1 Experimental Design

In this section we show the experimental setups. First, we present the simple battery based setup in Subsection 6.1.1. Then we show the setup in Subsection 6.1.2, in which a DSM based model is applied.

6.1.1 Simple Battery Setup

In this simple setup, we assume that there are no power outages (whether scheduled or unscheduled). Also, the design/model is unintelligent, as it does not have a lot of predicates and functions or we could say it cannot distinguish between peak and off-peak hours. Further, it simply has one charging action. This simple setup represents the case in which neither there is unreliability in the grid nor a DSM based strategy is used. By the way, the load profile is the same. The TIF values are inserted into the problem file as shown in Figure 6.1.

```
(at 18.0 (= (battery_soc) 70))

(at 19.0 (= (battery_soc) 63.95))
(at 20.0 (= (battery_soc) 53.47))
(at 21.0 (= (battery_soc) 34.68))
(at 22.0 (= (battery_soc) 21.19))
(at 23.0 (= (battery_soc) 10.36))

(at 24(day_ended))
```

Figure 6.1: Excerpt from the Problem File in Subsection 6.1.1

Since the power outages are not there the battery's SOC does not drop much and the system needs three charging actions to remain inside the limits. The planner output is shown in Figure 6.2.

The detailed planner output displays in Figure 6.3 that it took 0.074 seconds to find a plan in this simple battery based setup, while in comparison to the results we show in Subsection 6.2.1 and 6.2.2 its quite less. This is because they take 0.385 and 0.373 seconds to find plans. Also, the size of the state space to search for a solution is small, this simple case has a state space of 85 states, while the one we present in Subsection 6.2.1 has 440 states and the one in Subsection 6.2.2 with unscheduled power outages has 655. This is because this is a simple setup without any restrictions

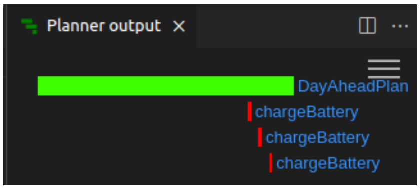


Figure 6.2: Planner Output of Subsection 6.1.1

and checks, and the planner is free to charge whenever it wants. As it does not consider important parameters such as Time of Unit (TOU) based times and power outages so it would be much costlier and less reliable for the consumers.

```
; States evaluated: 85
; Cost: 24.001
; Time 0.03
0.000: (dayaheadplan)  [24.001]
19.671: (chargebattery)  [0.330]
20.671: (chargebattery)  [0.330]
21.671: (chargebattery)  [0.330]
Planner found 1 plan(s) in 0.074secs.
```

Figure 6.3: Detailed Planner Output of Subsection 6.1.1

6.1.2 DSM based Setup

Here we present the DSM based setup, which at least has the TOU based timing involved in it. So it is intelligent enough to save costs, although still not able to cater for power outages. Also it does not have separate charging actions but we restrict the charging to off-peak time only. So this represents a simple DSM strategy application. This experimental design/setup also does not have the power outages rooted in it, thus it shows a case in which an unreliable power grid is not catered but DSM strategy is. The problem file is almost similar to the previous case with the exception that now peak and off-peak periods are defined as shown in Figure 6.4.

```
    (at 17.99 (not(off_peak))) ;PEAK started

    (at 19.0 (= (battery_soc) 63.95))
    (at 20.0 (= (battery_soc) 53.47))
    (at 21.0 (= (battery_soc) 34.68))
    (at 22.0 (= (battery_soc) 21.19))
    (at 23.0 (= (battery_soc) 10.36))

    (at 23.01 (off_peak)) ;PEAK ended
```

Figure 6.4: Excerpt from the Problem File in Subsection 6.1.2

The planner output has a similar number of actions as in Subsection 6.1.1, because all other variables have remained the same. The Planner output is shown in Figure 6.5. Only the difference is that the TOU based timing is applied in it, which effects on the scheduling times of the actions.

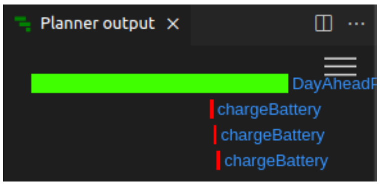


Figure 6.5: Planner Output of Subsection 6.1.2

As discussed in the above paragraphs, the scheduling times of the actions would be different than the previous case. This is clearly evident from Figure 6.6. Now, the AI planner restricts the charging actions to be applied before the peak time starts at 18:00 hours. Previously, they were applied at 19, 20 and 21 hours. But now the cost saving strategy is in place.

The detailed planner output in Figure 6.6 shows that this solution has a bit more states than the simple battery setup in the previous subsection. This is because it has constraints on the charging times and the AI planner has to enhance the search space in order to look for a solution. Though, it is still fairly less when we compare it with the result of scenarios that we present in Subsection 6.2.1 or Subsection 6.2.2, which has many restrictions and accommodates for both unreliability in the grid (as the first priority) and cost saving (as the second priority).

```
; States evaluated: 102
; Cost: 24.001
; Time 0.03
0.000: (dayaheadplan)  [24.001]
16.671: (chargebattery)  [0.330]
17.002: (chargebattery)  [0.330]
17.333: (chargebattery)  [0.330]
Planner found 1 plan(s) in 0.075secs.
```

Figure 6.6: Detailed Planner Output of Subsection 6.1.2

6.2 Results

We present the results in several scenarios. We define the standard peak hours from 18:00 to 23:00. Although, in two of the test cases (Subsection 6.2.6 and 6.2.7) we analyze the model with a 4 hour peak time period.

One scheme of scheduled power outage (or load shedding) is taken for simplicity and equality as a standard among the different scenarios. Though, it would be extended in the some of the test cases. The pre-defined standard scheduled power outages occur at the following hours 04:00 to 05:00, 09:00 to 10:00 and 15:00 to 16:00. While in some scenarios, we introduce more power outages that take place during and after the peak hours. In the first scenario no unscheduled power outage is anticipated by the predictor.

6.2.1 Scheduled power outage of 3 hours with 5 hour peak time period

This scenario has a three hour power outage at the standard predefined hours, I-e; at hours 04:00 - 05:00, 09:00 - 10:00 and 15:00 - 16:00. Plus the peak time period was also standard, I-e; from hours 18:00 - 23:00, 5 hours a day. While the load profile stays the same as comparison of different test cases has to be done, so this load profile which is one of the input to the system is kept fixed.

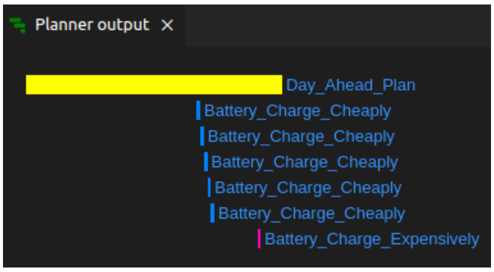


Figure 6.7: Planner Output of Subsection 6.2.1

AI planner makes a plan in which five cheap charging actions are applied, and one expensive charging action is applied also, can be seen in Figure 6.7. AI planner anticipates that the battery's SOC will fall below the lower limit during the peak hours, although it has exhausted the cheap charging options, but it still needs one more charge to satisfy the constraint. It could not have charged cheaply as then it would have gone above 100% SOC which is prevented by a precondition in the action and is practically not possible in real world. This is how it tries to save cost (DSM) as much as it can, plus provide uninterrupted power supply to the house.

```
; States evaluated: 440
; Cost: 24.001
; Time 0.27
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_cheaply)  [0.330]
16.331: (battery_charge_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
21.671: (battery_charge_expensively)  [0.330]
Planner found 1 plan(s) in 0.358secs.
```

Figure 6.8: Detailed Planner Output of Subsection 6.2.1

The detailed output from the AI planner is presented in Figure 6.8. It displays the exact times on which the actions were applied, and the duration of those actions. It tells that to find this solution the planner searched through 440 states, and took 0.358 seconds, then found one plan. The cost shown is just the time taken for the whole plan to complete (24 hours), it is not the energy cost. By analyzing the output, we can say that the planner does not charge during the hours on which the scheduled power outages take place. For instance, it seems that it wanted to charge somewhere around 15:00 hours but could not as the grid was not available (modeled by predicate "is_not_blackout"), and starts charging exactly at 16:00 hours when the grid gets back on. Also, it tries to fill up the battery as much as it can before the peak time period starts when electricity would get expensive, that is why it performs numerous cheap charging actions sometime before 18:00 hours.

6.2.2 Unscheduled power outage of 1 hour and Scheduled power outage of 3 hours

This test case has an unscheduled power outage of 1 hour from 13:00 to 14:00 hours, this was forecasted by the predictor that depends upon the exogenous weather variables to predict the random or unscheduled power outages for the next day in advance. Further, this case has the regular three hour scheduled power outage at the standard predefined hours, I-e; at hours 04:00 - 05:00, 09:00 - 10:00 and 15:00 - 16:00. Moreover the peak-time is also standard, I-e; from hours 18:00 - 23:00, 5 hours a day. While the load profile is kept same that is one of the inputs to the system.

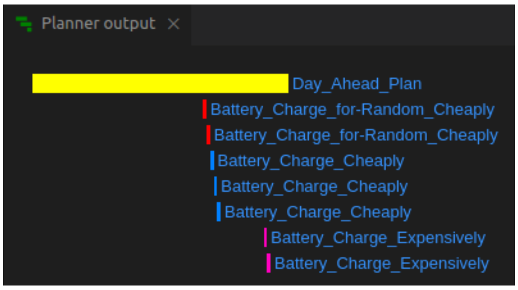


Figure 6.9: Planner Output of Subsection 6.2.2

It can be seen from Figure 6.9 that the AI planner makes a plan in which two preemptive cheap charging actions are applied, three cheap chagrining actions are also applied and two expensive charging actions are applied as well. The action "Battery_Charge_for-Random_Cheaply" is triggered by the unscheduled power outage of 1 hour. AI planner anticipates that the battery's SOC will fall below the lower limit during the peak hours, although it has consumed the cheap charging options, but it still needs two more charge to satisfy the constraint.

The AI planner did not charge cheaply earlier because boundary constraint was not in expected to cross the lower limit and also the top limit of 100% exists, so the AI planner acted when it believed that the boundaries will be breached. So, generally the AI planner tries to charge with the cheap actions whether caused by scheduled or unscheduled power outages. Thus, reducing energy cost. Also it shifts the load of the smart home to cheaply charged battery energy when the electricity tariff gets high at the daily peak hours. Most importantly, the system makes sure that there is no loss of power supply even if random power interruption occurs.

```
; States evaluated: 655
; Cost: 24.001
; Time 0.33
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
21.671: (battery_charge_expensively)  [0.330]
22.002: (battery_charge_expensively)  [0.330]
Planner found 1 plan(s) in 0.373secs.
```

Figure 6.10: Detailed Planner Output of Subsection 6.2.2

The Figure 6.10 shows the detailed output from the AI planner. It displays the exact times on which the actions were applied, and the duration of those actions. It found this solution in 0.373 seconds and searched through 655 states. Hence, it took more time and states in comparison to the previous scenario, this is because this case is a bit complex.

By analyzing the output, we can say that the planner does not charge during the hours on which the scheduled or unscheduled power outages occur, as it is prevented because of the modeling of predicates “is_not_random_blackout” and “is_not_blackout”. The AI planner tries to charge the battery as much as it can before the peak time period starts when electricity would get costly, that is why it performs numerous cheap charging actions sometime before 18:00 hours.

In comparison to the detailed output of the previous case in which only the scheduled power outages were considered, we have one more expensive charging action at around 22:00 hours. This is because the SOC decreased more in this scenario and the constraints and issues increased. For instance, the unscheduled power outage caused the SOC to decrease at 13:00 to 14:00 hours, meanwhile the charging window got reduced and the AI planner had to execute one more expensive charge. It did that to provide uninterrupted power to the smart home, as it becomes the first priority in this type of scenario, and the cost reduction gets demoted to the second priority.

Scenario 2 of Section 6.2.2

This case has same conditions, except change in peak hour time period. The peak hour time period in this case is 4.5 hours instead of 5 hours. This means the peak hour starts at 18:00 hours and ends at 22:30 hours, instead of 23:00 hours.

The planner output is displayed in Figure 6.11 we can see that after the expensive charging action, a cheap charging action is executed. In total, now we have six cheap charging actions and one expensive charging action.

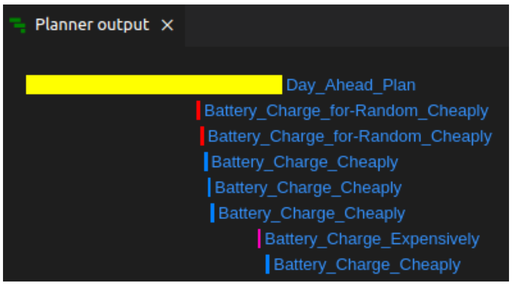


Figure 6.11: Planner Output of Subsubsection 6.2.2

In comparison to the previous case in Subsection 6.2.2, we now have one expensive charging action. This is because the AI planner intelligently shifted the last charging action. It postponed and pushed the last charging action after the peak time period (that is; at 22:30 hours) in order to reduce the electricity cost, see Figure 6.12. Thus, neither did it allowed the battery’s SOC to go under the lower bound of the mixed metric constraint nor it forgot to apply smart cost-saving DSM strategy that is embedded in the AI planning domain.

```
; States evaluated: 655
; Cost: 24.001
; Time 0.35
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
21.671: (battery_charge_expensively)  [0.330]
22.500: (battery_charge_cheaply)  [0.330]
Planner found 1 plan(s) in 0.405secs.
```

Figure 6.12: Detailed Planner Output of Subsubsection 6.2.2

6.2.3 Unscheduled power outage of 1 hour and Scheduled power outage of 4 hours, with 1 hour scheduled outage between 5 hour peak time period

This scenario was designed to have 1 hour unscheduled power outage as in previous case and three hours of scheduled power outages at the standard pre-defined hours already mentioned in earlier. Along with that one more scheduled power outage during the peak time period which occurs at 21:00 hours to 22:00 hours.

The excerpt from the modified problem file is shown in Figure 6.13, the power outage occurs at 21:00 hours and the power is restored from the utility by 22:00 hours, the power outage is represented by the false and true assertion of "is_not_blackout" predicate. While, the peak time period was standard as defined in Section 6.2. The load profile stays the same as comparison of different test cases has to be done.

```
(at 20.0 (= (battery_soc)24.04080833333333))
(at 21.0 (= (battery_soc)5.24776666666666))

(at 21.0 (not(is_not_blackout))) ;SCHEDULED POWER OUTAGE
(at 22.0 (is_not_blackout))

(at 22.0 (= (battery_soc)-9.307875000000008))
(at 23.0 (= (battery_soc)-20.139233333333344))
```

Figure 6.13: Portion of the Problem File modified in Subsection 6.2.3

The planner output in the colorful user-friendly form is presented by Microsoft VS Code, can be seen in Figure 6.14. It has the same no of actions as in the previous case in Subsection 6.2.2, because the extra one hour power outage between the peak time period did not matter as the peak hours are already considered as significant hours and planner takes them into consideration for DSM strategy.

The only difference is in the scheduling times of the Battery_Charge_Expensively actions. What happens is that the first expensive charging action now starts at 22:00 hours just when the power from the grid is back, see Figure 6.15. In the experiment analyzed in Subsection 6.2.2, the first expensive charging action started almost 40 minutes past 21:00. So the AI planner logically shifted the charging action a little further as it is not allowed to charge during a power outage, because of the false assertion of predicate "is_not_blackout" from 21:00 to 22:00 hours.

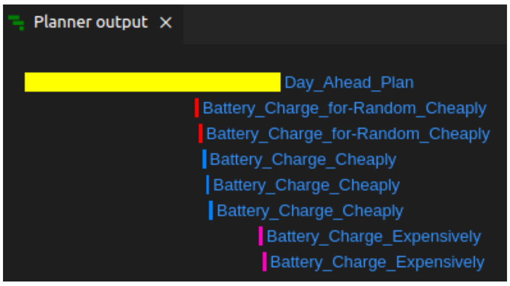


Figure 6.14: Planner Output of Subsection 6.2.3

```
; States evaluated: 654
; Cost: 24.001
; Time 0.33
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
22.000: (battery_charge_expensively)  [0.330]
22.331: (battery_charge_expensively)  [0.330]
Planner found 1 plan(s) in 0.407secs.
```

Figure 6.15: Detailed Planner Output of Subsection 6.2.3

6.2.4 Scheduled power outage of 3 hours and Unscheduled power outage of 2 hours and with 1 hour unscheduled outage between 5 hour peak time period

This test case has three hours of scheduled power outages at the standard pre-defined hours, plus it has 2 hours of unscheduled power outages. The first unscheduled power outage is at same 13:00 to 14:00 hours but the second unscheduled power outage predicted is from 22:00 to 23:00 hours.

A portion from the problem file of this case is shown in Figure 6.16, the second unscheduled power outage occurs at 22:00 hours and the power is restored from the grid by 22:00 hours, this power outage is represented by the false and true assertion of "is_not_random_blackout" predicate. The peak time period was same as defined in Section 6.2. The Timed Initial Literal (TIL) named "random_shed_during_peak" gets activated in advance by the problem instance of the AI planning. That is done to intimate the AI planning domain instance of the random blackout ahead, so that it can handle cater it proactively.

```
(at 21.0 (= (battery_soc)5.24776666666666))

(at 21.0 (random_shed_during_peak)) ; RANDOM POWER OUTAGE COMING AHEAD
(at 22.0 (= (battery_soc)-9.307875000000008))
(at 22.0 (not(is_not_random_blackout))) ; RANDOM POWER OUTAGE
(at 23.0 (is_not_random_blackout))

(at 23.0 (= (battery_soc)-20.139233333333344))
```

Figure 6.16: Portion of the Problem File modified in Subsection 6.2.4

The planner output in Figure 6.17 shows that three types of durative charging actions were performed to achieve the solution in this specific problem. The first two actions "Battery_Charge_for-Random_Cheaply" were executed because of the unscheduled power outage that occurred at 13:00 hours. After that, the AI planner charges the battery in advance considering the long peak hours in which it had to shift the load of the smart home to the battery in order to save cost and implement the DSM strategy. The second last action "Battery_Charge_for-Random_Expensively" ran because the system had the knowledge about the second unscheduled power outage which would occur at 22:00 hours, so it charged up in order to avoid loss of power supply to the residential unit.

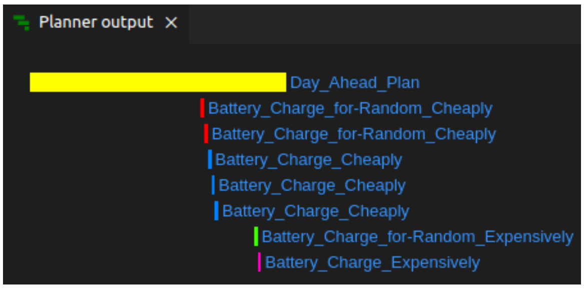


Figure 6.17: Planner Output of Subsection 6.2.4

In the detailed output of this case shown in Figure 6.18 it is visible that the assertion of predicate "random_shed_during_peak" at 21:00 hours sent a signal to the AI planner that the lower bound of the battery's SOC might get violated. So the AI planner in an intelligent way charged the battery pre-emptively at 21:01 using the "Battery_Charge_for-Random_Expensively" action, because it in this type of scenario the priority is to provide uninterrupted power so it charged during peak hours.

```
; States evaluated: 660
; Cost: 24.001
; Time 0.38
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
21.001: (battery_charge_for-random_expensively)  [0.330]
21.332: (battery_charge_expensively)  [0.330]
Planner found 1 plan(s) in 0.441secs.
```

Figure 6.18: Detailed Planner Output of Subsection 6.2.4

6.2.5 Scheduled power outage of 3 hours and Unscheduled power outage of 3 hours and with 1 hour unscheduled outage between 5 hour peak time period

This test case has three hours of scheduled power outages at the standard pre-defined hours, plus it has three hours of unscheduled power outages as well. The first unscheduled power outage is at same 13:00 to 14:00 hours, the second unscheduled power outage predicted is from 22:00 to 23:00 hours and the third from 23:00 to 24:00 hours.

In the Figure 6.19 the extract from the problem file is shown, the second unscheduled power outage occurs at 22:00 hours and the third at 23:00 till 24:00 hours, this power outage is represented by the false and true assertion of “is_not_random_blackout” predicate twice in the problem file. The peak time period was same as defined in Section 6.2. The TIL named “random_shed_during_peak” gets asserted in advance by the problem instance of the AI planning. That is done to inform the AI planning domain instance of the random blackouts coming ahead, so that it can handle cater them proactively.

```
(at 21.0 (random_shed_during_peak)) ; RANDOM POWER OUTAGE COMING AHEAD
(at 22.0 (= (battery_soc)-9.307875000000008))
(at 22.0 (not(is_not_random_blackout))) ; RANDOM POWER OUTAGE
(at 23.0 (is_not_random_blackout))

(at 23.0 (= (battery_soc)-20.139233333333344))
(at 23.01 (not(peak)))
(at 23.02 (off_peak))

(at 23.3 (not(is_not_random_blackout))) ; RANDOM POWER OUTAGE
(at 24.0 (is_not_random_blackout))
(at 24.0 (= (battery_soc)-30.332525000000015))
```

Figure 6.19: Portion of the Problem File modified in Subsection 6.2.5

The planner output displayed in Figure 6.20 shows that four types of durative charging actions were performed to achieve the solution in this problem. The first two actions “Battery_Charge_for-Random_Cheaply” were executed because of the unscheduled power outage that occurred at 13:00 hours. Then it returned to normal mode to charges the battery in advance considering the long peak hours in which it had to shift the load of the smart home to the battery in order to save cost and implement the DSM strategy. But during the end of peak hours the system predicted that two hours of unscheduled power outages will take place. So the AI planner primarily charged the battery using “Battery_Charge_for_Random_Expensively” actions twice as they were activated by the corresponding predicate. And when the emergency (unscheduled power outages) were handled then it applied the “Battery_Charge_Expensively”, which it had to earlier, but its priority got reduced because of the unscheduled power outages forecasted by the predictor.

It does this because it anticipates power outages by the help of TIFs, and in order to provide uninterrupted power to the residential unit it charges expensively more. Because the highest priority is to provide electricity at all times, then comes the energy saving aspect.

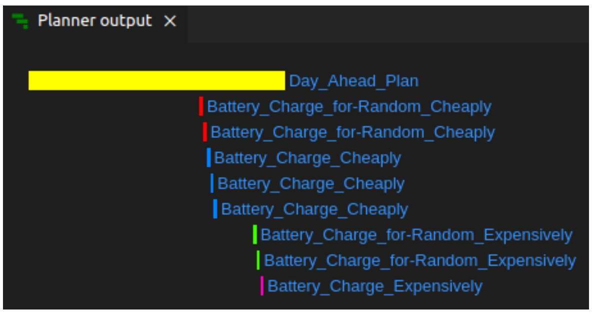


Figure 6.20: Planner Output of Subsection 6.2.5

In the detailed output of this case shown in Figure 6.21 it is clear that the assertion of predicate "random_shed_during_peak" at 21:00 hours sent a warning sign to the AI planner that the lower bound of the battery's SOC might get violated. In response, the AI planner in an intelligent way charged the battery pre-emptively during 21:00 hours twice using the "Battery_Charge_for-Random_Expensively" action, because it in this type of scenario the priority is to provide uninterrupted power so it charged during peak hours. The AI planner did not had much SOC because of the long peak hours (5 hours), plus 6 hours of scheduled and unscheduled power outages, that is why it tried to charge the battery cheaply before the peak hours, but when it had no other choice then its priority changed and it charged using expensive charging actions. Though, first it used the action triggered by the unscheduled interruption, then switched to the other one.

```
; States evaluated: 670
; Cost: 24.001
; Time 0.45
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
21.001: (battery_charge_for-random_expensively)  [0.330]
21.332: (battery_charge_for-random_expensively)  [0.330]
21.663: (battery_charge_expensively)  [0.330]
Planner found 1 plan(s) in 0.524secs.
```

Figure 6.21: Detailed Planner Output of Subsection 6.2.5

6.2.6 Scheduled power outage of 3 hours, plus 2 hours of Unscheduled power outage with one outage sometime after 4 hour peak time period

The predefined standard power outages occur at the same times as in the cases above, that is; 04:00 to 05:00, 09:00 to 10:00 and 15:00 to 16:00. Further, there is a one hour unscheduled power outage from 13:00 till 14:00 hours. These power outages are kept same during the experiments so that the comparison of different test cases is performed easily. Apart from that, it is predicted that there will be a one hour unscheduled power outage at 23:00 hours till 24:00 as shown in Figure 6.22, that is; one hour after the peak time period ends. The TIF at time 22.0 hours in the problem file increases the value of the function "random_run_capacity_value" so the AI planner can run additional cheap durative actions. The peak hours in this case and the next one are from 18:00 to 22:00, that is; 4 hours peak time period.

Five cheap charging actions are applied by the AI planner before the peak hours, the first two are caused because of the unscheduled power interruptions. While the other two because of scheduled power outages plus preemptive peak hour energy harvesting. As given in the TIF that there would be a random power outage at 23:00 hours. So the planner has to do something regarding that. It performs additional charging actions as shown in Figure 6.23.

The charging action applied at the end is performed just when the peak time ends at 22 hours as shown in Figure 6.24, because the secondary priority kicks in and the AI planner schedules cheap charging actions. The actions named "Battery_Charge_for-Random_Cheaply" are applied in this case because of the predicted unscheduled power interruption. In comparison to the experimental result in Subsection 6.2.2, the 6th and 7th charging actions are cheap charging actions, instead of an

```
(at 17.98 (not(off_peak)))
(at 17.99 (peak))
(at 19.0 (= (battery_soc)34.516191666666664))
(at 20.0 (= (battery_soc)24.040808333333333))
(at 21.0 (= (battery_soc)5.247766666666666))
(at 22.0 (= (battery_soc)-9.307875000000008))

(at 22.01 (not(peak)))
(at 22.02 (off_peak))
(at 22.0 (= (random_run_capacity_value)4))

(at 23.0 (not(is_not_random_blackout))) ; RANDOM POWER OUTAGE
(at 24.0 (is_not_random_blackout))
(at 24.0 (= (battery_soc)-20.139233333333344))

(at 24 (day_ended))
```

Figure 6.22: Portion of the Problem File modified in Subsection 6.2.6

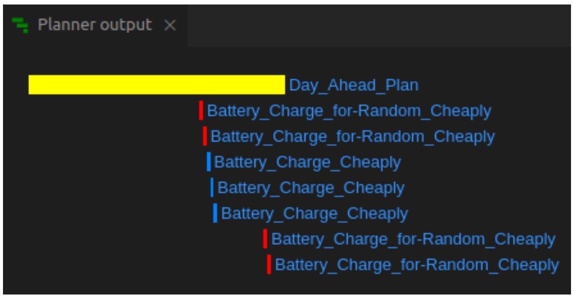


Figure 6.23: Planner Output of Subsection 6.2.6

expensive charging action in the case presented in Subsection 6.2.2. This is because the battery was not much reduced because of the 4 hour peak time period and also the AI planner knew that it can charge cheaply as it has one hour gap between the peak ending time (22:00 hours) and the upcoming power outage (23:00 hours).

```
; States evaluated: 655
; Cost: 24.001
; Time 0.35
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
22.020: (battery_charge_for-random_cheaply)  [0.330]
22.351: (battery_charge_for-random_cheaply)  [0.330]
Planner found 1 plan(s) in 0.401secs.
```

Figure 6.24: Detailed Planner Output of Subsection 6.2.6

6.2.7 Scheduled power outage of 3 hours, plus 2 hours of Unscheduled power outage with one outage just after 4 hour peak time period

The standard scheduled power outages occur at the same times as in the cases above. Furthermore, there is a one hour unscheduled power outage from 13:00 till 14:00 hours. These power outages are kept same during the experiments so that the comparison of different test cases is performed easily. Apart from that, it is predicted that there is a one hour unscheduled power outage at 22:00 hours till 23:00 as shown in Figure 6.25, that is; just after the peak time period ends. The TIL at time 21.0 hours in the problem file asserts the relevant predicate so the AI planner can proactively run expensive preemptive charging actions which are triggered due to unscheduled power interruption. The peak hours in this case are from 18:00 to 22:00, that is; 4 hours peak time period.

```
(at 17.98 (not(off_peak)))
(at 17.99 (peak))
(at 19.0 (= (battery_soc)34.516191666666664))
(at 20.0 (= (battery_soc)24.04080833333333))
(at 21.0 (= (battery_soc)5.24776666666666))
(at 22.0 (= (battery_soc)-9.307875000000008))

(at 21.0 (random_shed_during_peak)) ; RANDOM POWER OUTAGE COMING AHEAD
(at 22.01 (not(peak)))
(at 22.02 (off_peak))

(at 22.0 (not(is_not_random_blackout))) ; RANDOM POWER OUTAGE
(at 23.0 (is_not_random_blackout))
(at 23.0 (= (battery_soc)-20.139233333333344))

(at 24 (day_ended))
```

Figure 6.25: Portion of the Problem File modified in Subsection 6.2.7

Five cheap charging actions are applied by the AI planner before the peak hours. A TIL alongside a TIF at 22:00 hours tells the AI planner that there would be a power outage at 22:00 hours. So the planner performs expensive charging actions (Battery_Charge_for-Random_Expensively) that were activated because of unscheduled power outage as shown in Figure 6.26. This is because it estimates that if it does not do that then the constraint might not be satisfied. This is how the AI planner intelligently provides uninterrupted power and sidelines the cost saving during this. In comparison to the previous in Subsection 6.2.6, there is no time in between the peak ending and the power outage, so the AI planner has to act before that.

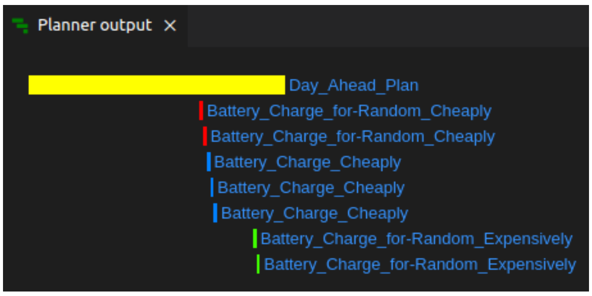


Figure 6.26: Planner Output of Subsection 6.2.7

In the detailed planner output shown in Figure 6.27, it can be seen that the planner applies the expensive charging actions at 21.001 and 21.332 hours. The AI planner could not apply cheap actions because it was restricted by DSM strategy (peak hours).

```
; States evaluated: 656
; Cost: 24.001
; Time 0.43
0.000: (day_ahead_plan)  [24.001]
16.000: (battery_charge_for-random_cheaply)  [0.330]
16.331: (battery_charge_for-random_cheaply)  [0.330]
16.662: (battery_charge_cheaply)  [0.330]
16.993: (battery_charge_cheaply)  [0.330]
17.324: (battery_charge_cheaply)  [0.330]
21.001: (battery_charge_for-random_expensively)  [0.330]
21.332: (battery_charge_for-random_expensively)  [0.330]
Planner found 1 plan(s) in 0.523secs.
```

Figure 6.27: Detailed Planner Output of Subsection 6.2.7

6.3 Discussion

We conclude the Chapter 6 and discuss the results together with experimental design/setup in this section. We present the comparison between Baseline (load profile), only Scheduled power outage model, and Unscheduled plus Scheduled power outage model.

The graph of power drawn (y-axis) versus time of the day (x-axis) is shown in the Figure 6.28. The unit of power is kWh, while the unit for time is hours. The total power drawn by the house at each hour is shown in the graph. The blue line represents the baseline that is the actual load profile of a day from the PRECON dataset [NA19] on which we have analyzed our model. The red dotted line shows the power drawn by the house when our DSM along with scheduled power outage handling model was applied. The green line shows the power drawn when our DSM together with unscheduled and scheduled power outage controlling model was applied.

It can be clearly observed that during the peak-hours (from 18:00 to 23:00 hours) the energy used is minimal when the AI planner based model is implemented. The experimental scenario performed in Subsection 6.2.1 and Subsection 6.2.2 was considered for making this graph. The Subsection 6.2.1 represents the red dotted curve (DSM with scheduled interruptions only), while the Subsection 6.2.2 shows the green curve (DSM with unscheduled and scheduled interruptions). In the red curve, only once the AI planner asks the power inverter to charge batteries in between the peak hours, this happens at 21:00 hours. In the green curve, the AI planner signals the power inverter twice to charge batteries between the peak hours.

Here, we present statistics, the total off-peak time consumption in the baseline (load profile) for the day is 30.52 KWH, and the total peak time consumption is 12.14 KWH. In our model having DSM with unscheduled and scheduled power outages, the total off-peak consumption is 39.21 KWH, while the total off-peak consumption is 3.46 KWH. In our proposed third type, the model having with DSM with only scheduled power outages, the total off-peak consumption is 39.21 KWH, while the total off-peak consumption is 1.73 KWH.

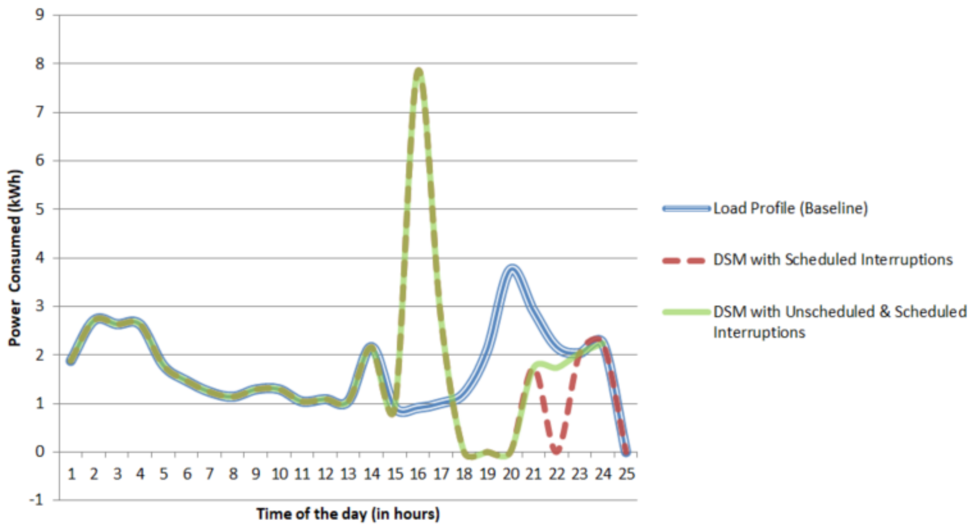


Figure 6.28: Graph based comparison between Baseline, only Scheduled outage model, and Unscheduled plus Scheduled outage model

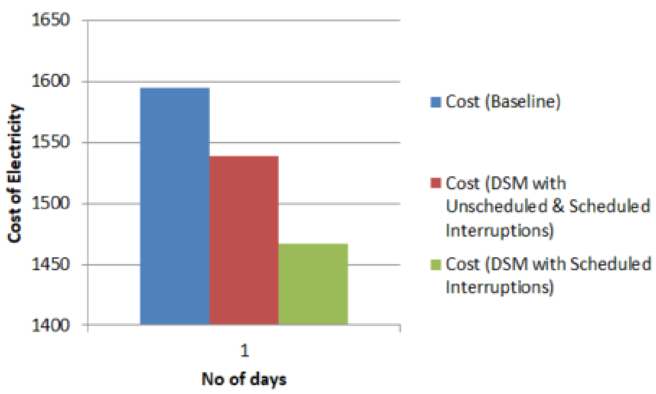


Figure 6.29: Cost of Electricity for Baseline, only Scheduled outage model, and Unscheduled plus Scheduled outage model

The cost of electricity for the whole day considering the three types of models is shown in the Figure 6.29. The blue bar represents the baseline model (model no. 1) cost for one complete day, the red bar shows the cost when DSM alongside unscheduled and scheduled power outage management model (model no. 2) is enforced, the green bar displays the cost when DSM along with only scheduled power outage handling model (model no. 3) is applied.

The load profile (baseline) which does not have any mechanism to counter the unreliable grid conditions has the highest cost of electricity, meaning if a scheduled or unscheduled power outage occurs the residential unit will experience blackout and the user's comfort level will be lowered severely. Even if the user installs an automatic gasoline generator then its overall energy cost increases because of the fuel costs and maintenance. On the other hand, if user has an unintelligent uninterrupted power supply device, then energy cost will increase since the device is not clever enough and could charge during peak hours. It does not work proactively, that is why the risk of blackout still remains in this case. Hence, no other easily available approach can accommodate for scheduled and most importantly unscheduled power outages to provide solution for this specific problem.

In the result of our models, there is a considerably big decline in the peak time consumption of about 10 KWH, which is reflected in the off-peak consumption as it has climbed up to 39.21 KWH from 30.52 KWH, see Figure 6.28. This is because most charging actions are performed during off-peak hours in order to provide power during the significant hours. Although, the important point is that this model not only saves cost but provides uninterrupted power during all times in an environment where the power grid is unreliable. From the statistics it seems that the power consumption has increased, but the energy cost depends on the time of use (TOU) tariffs, as there are different rates for peak and off-peak times. So the total energy cost is the important factor that should be analyzed to find out the most efficient model.

IESCO provides two rates for electricity, 41.89 Rupees (local currency) for peak and 35.57 Rupees for off-peak [Isld]. Hence, we calculated the electricity cost using the tariff and plotted the graph in Figure 6.29. When the model no.3 is implemented by our AI planner then we pay 8% less cost of electricity in comparison to (model no.1). When DSM alongside unscheduled and scheduled power outage management model (model no.2) is implemented by our AI planner then we pay 3.5% less cost of electricity in comparison to (model no.1). Along with the energy cost savings presented earlier, our system guarantees uninterrupted power supply to the smart home. This is the salient feature of our model, since the power grid is unreliable.

7 Conclusion and Outlook

7.1 Conclusion

The thesis has investigated the suitability of a hybrid approach for demand-side management in unreliable power grids. The evidence from this thesis suggests that hybrid approach is effective for handling the complex nature of unreliable power grids. Temporal Planning has proven to be a successful technique to shift the load of smart home from peak to off-peak tariff hours, thus, saving energy cost by implementing DSM strategy. Additionally, it solves the issue of scheduled power outages, which are common in developing nations' power networks.

LSTM based deep learning model has impressively resolved the prediction problem of unscheduled power outages. Several exogenous variables taken from the day ahead weather forecast are inserted into the model to predict the non-linear random power interruptions. Hence, combining Temporal Planning and LSTM model completely addresses our problem statement. The day-ahead plans generated in our work intelligently tries to save electricity cost, while also making sure that the power supply is not interrupted. Therefore, protecting the smart home from the negative consequences of an unreliable power grid.

7.2 Outlook

The United Kingdom has the highest amount of offshore wind capacity globally, but the inconsistency of wind speeds remains a significant challenge [UMA+22]. To address this issue, they are building a battery storage facility as it is necessary to store the electricity produced by wind and solar power sources. Similarly, Økland states that in the coming years, the energy system will witness a significant increase in the adoption of renewable energy sources, resulting in more variable electricity generation due to the reliance on weather-dependent intermittent renewable energy sources [Økl22]. The performance of the future power system is put to the test by the growth of solar power and flexible loads, which result in large frequency fluctuations [WWT+19]. This necessitates greater adaptability in the energy system, both in terms of supply and demand.

For this reason, we believe that employing AI planning alongside machine learning methods in this domain would be advantageous in the future. As most countries are shifting towards renewable sources of energy. We need much better energy management systems powered by intelligence, since the renewable sources do not supply a fixed amount of energy at all times, it is continuously changing. Thus, this area needs further research to investigate the reliable grids of developed countries in scenarios when their renewable energy output expands.

Bibliography

[A G] A. Green and B. J. Reji and C. Muise and E. Scala and F. Meneguzzi and F. M. Rico and H. Stairs and J. Dolejsi and M. Magnaguagno and J. Mounty. *The AI Planning and PDDL Wiki*. URL: https://planning.wiki/ (cit. on p. 27).

[AA18] G. Azkune, A. Almeida. "A Scalable Hybrid Activity Recognition Approach for Intelligent Environments". In: *IEEE Access* (2018). DOI: https://doi.org/10.1109/ACCESS.2018.2861004 (cit. on pp. 18, 42).

[AAAF22] M.-T. E. Astal, A. Alhabbash, A. Abu-Hudrouss, G. Frey. "Deep Learning-based Power Load Shedding Approach for Gaza's Electricity Grid". In: *IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (EEEIC / ICPS Europe)* (2022). DOI: https://doi.org/10.1109/EEEIC/ICPSEurope54979.2022.9854713 (cit. on pp. 32, 34, 38, 49).

[AALC15] G. Azkune, A. Almeida, D. López-de-Ipiña, L. Chen. "Extending knowledge-driven activity models through data-driven learning techniques". In: *Expert Systems with Applications* (2015). DOI: https://doi.org/10.1016/j.eswa.2014.11.063 (cit. on pp. 17, 41, 42).

[AE07] M. H. Albadi, E. F. El-Saadany. "Demand Response in Electricity Markets: An Overview". In: *In Proceedings of the 2007 IEEE power engineering society general meeting* (2007). DOI: https://doi.org/10.1109/PES.2007.385728 (cit. on p. 22).

[AFG21] M. Aiello, L. Fiorini, I. Georgievski. "Software Engineering Smart Energy Systems". In: *Handbook of Smart Energy Systems*. Ed. by M. Fathi, E. Zio, P. M. Pardalos. Springer International Publishing, 2021, pp. 1–29. ISBN: 978-3-030-72322-4. DOI: 10.1007/978-3-030-72322-4_21-1. URL: https://doi.org/10.1007/978-3-030-72322-4_21-1 (cit. on p. 44).

[Baj16] J. Bajada. "Temporal Planning for Rich Numeric Contexts". PhD thesis. University of London, 2016 (cit. on pp. 26, 44).

[Bal] Balkan Green Energy News. *Peak and Off-Peak timing in North Macedonia*. URL: https://balkangreenenergynews.com/north-macedonia-to-reinstate-cheaper-afternoon-power-tariff-for-households/ (cit. on p. 46).

[CCFL15] A. Coles, A. Coles, M. Fox, D. Long. "Forward-Chaining Partial-Order Planning". In: *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS 2010)*. 2015 (cit. on p. 19).

[CCFL21] A. Coles, A. Coles, M. Fox, D. Long. "Forward-Chaining Partial-Order Planning". In: 20 (May 2021), pp. 42–49. DOI: https://doi.org/10.1609/icaps.v20i1.13403 (cit. on p. 66).

[DMB22] B. Dey, F. P. G. Márquez, A. Bhattacharya. “Demand side management as a mandatory inclusion for economic operation of rural and residential microgrid systems”. In: *Sustainable Energy Technologies and Assessments* (2022). DOI: https://doi.org/10.1016/j.seta.2022.102903 (cit. on p. 22).

[EH03] S. Edelkamp, J. Hoffmann. “PDDL2.2 The Language for the Classical Part of the the International Planning Competition”. In: (2003). DOI: https://planning.wiki/_citedpapers/pddl222004.pdf (cit. on pp. 18, 26).

[Enda] Endeavour Energy. *Current Planned and Unplanned Outages Map*. URL: https://www.endeavourenergy.com.au/outages/current-power-outages (cit. on p. 33).

[Endb] Endeavour Energy. *Power Outages, Unplanned Outages*. URL: https://www.endeavourenergy.com.au/outages/planned-unplanned-outages/power-outages (cit. on pp. 17, 34).

[Ene] Energy and Water Services Regulatory Commission of the Republic of North Macedonia. *Electricity Tariff in North Macedonia, Pricing, Peak / Off-Peak tariff*. URL: https://www.erc.org.mk/page_en.aspx?id=287 (cit. on p. 46).

[FN71] R. E. Fikes, N. J. Nilsson. “Strips: A new approach to the application of theorem proving to problem solving”. In: *Artificial Intelligence, 2nd International Joint Conference on Artificial Intelligence* (1971). DOI: https://doi.org/10.1016/0004-3702(71)90010-5 (cit. on p. 24).

[GA16] I. Georgievski, M. Aiello. “Automated Planning for Ubiquitous Computing”. In: *ACM Computing Surveys* (2016). DOI: https://doi.org/10.1109/TC.2016.2538237 (cit. on p. 24).

[GHL+09] A. E. Gerevini, P. Haslum, D. Long, A. Saetti, Y. Dimopoulos. “Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners”. In: *Artificial Intelligence* (2009). DOI: https://doi.org/10.1016/j.artint.2008.10.012 (cit. on p. 28).

[GSA23] I. Georgievski, M. Z. Shahid, M. Aiello. “AI temporal planning for energy smart buildings”. In: *Energy Informatics* 6.1 (Oct. 2023), p. 18. ISSN: 2520-8942. DOI: 10.1186/s42162-023-00289-w. URL: https://doi.org/10.1186/s42162-023-00289-w (cit. on pp. 18, 19, 43, 44).

[HLMM] P. Haslum, N. Lipovetzky, D. Magazzeni, C. Muise. *An Introduction to the Planning Domain Definition Language, Temporal Planinng*. Springer International Publishing (cit. on p. 25).

[Hom] Home-Energy Stats UK. *Energy-stats 2021*. URL: https://www.iea.org/reports/energy-access-outlook-2017 (cit. on p. 23).

[HPA+17] T. Hossen, S. J. Plathottam, R. K. Angamuthu, P. Ranganathan, H. Salehfar. “Short-term load forecasting using deep neural networks (DNN)”. In: *North American Power Symposium (NAPS)* (2017). DOI: https://doi.org/10.1109/NAPS.2017.8107271 (cit. on p. 36).

[HS97] S. Hochreiter, J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* (1997). DOI: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on pp. 40, 41).

[IM] G. Ilche, A. Marco. “Building Automation Based on Temporal Planning” (cit. on pp. 18, 25–27).

[Inta] International Energy Agency. *Energy Access Outlook 2017*. URL: https://www.iea.org/reports/energy-access-outlook-2017 (cit. on p. 34).

[Intb] International-U.S. Energy Information Administration (EIA). *EIA-India*. URL: https://www.eia.gov/international/overview/country/IND (cit. on p. 33).

[Isla] Islamabad Electric Supply Company. *IESCO Annual Performance Reports*. URL: https://www.iesco.com.pk/index.php/customer-services/dailymonthly-yearly-data (cit. on p. 33).

[Islb] Islamabad Electric Supply Company. *IESCO Maintenance Schedule*. URL: https://www.iesco.com.pk/index.php/customer-services/annual-maintenance-schedule (cit. on p. 33).

[Islc] Islamabad Electric Supply Company. *Load Shedding Management Schedule and Maintenance Schedule*. URL: https://www.iesco.com.pk/index.php/customer-services/loadshedding-mgt-schedule (cit. on p. 33).

[Isld] Islamabad Electric Supply Company. *Schedule of electricity Tariffs*. URL: https://iesco.com.pk/index.php/customer-services/tariff-guide (cit. on pp. 23, 46, 63, 90).

[Jus23] Jussi Rintanen. *A brief overview of temporal planning in AI*. 2023. URL: https://users.aalto.fi/~rintanj1/temporalplanning.html#:~:text=Temporal%20planning%20involves%20solving%20the,the%20actions%20and%20their%20effects. (cit. on p. 25).

[KLWK20] M. Khodayar, G. Liu, J. Wang, M. E. Khodayar. “Deep learning in power systems research: A review”. In: *CSEE Journal of Power and Energy Systems* (2020). DOI: https://doi.org/10.17775/CSEEJPES.2020.02700 (cit. on pp. 17, 35, 36, 41, 49).

[KMSM16] J. Khoury, R. Mbayed, G. Salloum, E. Monmasson. “Design and implementation of a real time demand side management under intermittent primary energy source conditions with a PV-Battery backup system”. In: *Energy and Buildings* 133 (Sept. 2016). DOI: 10.1016/j.enbuild.2016.09.036 (cit. on pp. 17, 46, 59).

[KR92] K. Kira, L. A. Rendell. “A Practical Approach to Feature Selection”. In: *Machine Learning Proceedings* (1992). DOI: https://doi.org/10.1016/B978-1-55860-247-2.50037-1 (cit. on pp. 38, 49).

[LNA17] G. Lemaître, F. Nogueira, C. K. Aridas. “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning”. In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: http://jmlr.org/papers/v18/16-365.html,%20https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html (cit. on p. 53).

[LPGA23] Y. Liu, L. Palmieri, I. Georgievski, M. Aiello. “Human-Flow-Aware Long-Term Mobile Robot Task Planning Based on Hierarchical Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* (2023). DOI: https://doi.org/10.1109/LRA.2023.3280816 (cit. on p. 42).

[LXR20] Q. Li, Y. Xu, C. Ren. “A Hierarchical Data-Driven Method for Event-Based Load Shedding Against Fault-Induced Delayed Voltage Recovery in Power Systems”. In: *IEEE Transactions on Industrial Informatics* (2020). DOI: https://doi.org/10.1109/TII.2020.2993807 (cit. on pp. 32, 37).

[M F03] D. L. M. Fox. “PDDL2. 1: An extension to PDDL for expressing temporal planning domains”. In: *Journal Of Artificial Intelligence Research* (2003). DOI: 10.1613/jair.1129 (cit. on pp. 24–26).

[MDG+19] J. Ma, Y. Ding, V. J. L. Gan, C. Lin, Z. Wan. “Spatiotemporal Prediction of PM2.5 Concentrations at Different Time Granularities Using IDW-BLSTM”. In: *IEEE Access* (2019). DOI: https://doi.org/10.1109/ACCESS.2019.2932445 (cit. on p. 40).

[MG23] U. S. Mirza, I. Georgievski. *Exploiting AI planning for Demand Side Management in Unreliable Power Grids*. 2023 (cit. on p. 17).

[MGH+98] D. McDermott, M. Ghallab, A. E. Howe, C. A. Knoblock, A. Ram, M. M. Veloso, D. S. Weld, D. E. Wilkins. “PDDL-the planning domain definition language”. In: 1998. URL: https://api.semanticscholar.org/CorpusID:59656859 (cit. on p. 24).

[Min] Ministry of Power, Government of India. *Power Sector at a Glance ALL INDIA*. URL: https://powermin.gov.in/en/content/power-sector-glance-all-india (cit. on p. 33).

[Mit97] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. DOI: https://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf (cit. on p. 31).

[Mon17] A. A. Monyei C.G. “Demand Side Management potentials for mitigating energy poverty in South Africa”. In: *Enegry Policy* (2017) (cit. on p. 33).

[MS08] A. I. Moustapha, R. R. Selmic. “Wireless Sensor Network Modeling Using Modified Recurrent Neural Networks: Application to Fault Detection”. In: *IEEE Transactions on Instrumentation and Measurement* (2008). DOI: https://doi.org/10.1109/TIM.2007.913803 (cit. on p. 39).

[MVC17] U. Minnaar, W. Visser, J. Crafford. “An economic model for the cost of electricity service interruption in South Africa”. In: *Utilities Policy* (2017) (cit. on p. 33).

[NA19] A. Nadeem, N. Arshad. “PRECON: Pakistan Residential Electricity Consumption Dataset”. In: *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. e-Energy ’19. ACM, 2019, pp. 52–57. DOI: 10.1145/3307772.3328317 (cit. on pp. 58, 88).

[NER] NERC 2022 State of Reliability. *How NERC Defines BPS Reliability*. URL: https://rb.gy/5xidxi (cit. on pp. 31, 32, 49).

[Nig21] Nigerian Electricity Regulatory Commission. *Third Quarter 2018 Nerc Quarterly Reports 2*. 2021. URL: https://nerc.gov.ng/ (cit. on p. 34).

[NLM15] I. E. Naqa, R. Li, M. J. Murphy. *Machine Learning in Radiation Oncology, What Is Machine Learning?* Springer Cham, 2015. ISBN: 978-3-319-18305-3. DOI: https://doi.org/10.1007/978-3-319-18305-3 (cit. on p. 31).

[Nor] North American Electric Reliability Corporation. *Standard EOP-003-1, Load Shedding Plans*. URL: https://rb.gy/lpnnak (cit. on p. 32).

[Økl22] G. L. Økland. *Demand Response in bottom-up planning models*. 2022 (cit. on p. 91).

[PAFL15] C. Piacentini, V. Alimisis, M. Fox, D. Long. “An extension of metric temporal planning with application to AC voltage control”. In: *Artificial Intelligence* (2015). DOI: https://doi.org/10.1016/j.artint.2015.08.010 (cit. on pp. 19, 26–28, 31, 43, 44, 64).

[PD11] P. Palensky, D. Dietrich. “Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads”. In: *IEEE Transactions on Industrial Informatics* (2011). DOI: https://doi.org/10.1109/TII.2011.2158841 (cit. on pp. 21–23).

[PFL+] W. Piotrowski, M. Fox, D. Long, D. Magazzeni, F. Mercorio. “Heuristic Planning for PDDL+ Domains”. In: *The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence Planning for Hybrid Systems: Technical Report WS-16-12* (cit. on p. 27).

[PFL15] C. Piacentini, M. Fox, D. Long. “Planning with numeric Timed Initial Fluents”. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015 (cit. on pp. 19, 27).

[Pla16] PlanningWiki. *What is ai planning*. 2016. URL: https://planning.wiki (cit. on p. 17).

[PMM09] G. D. Penna, D. Magazzeni, F. Mercorio. “UPMurphi: A Tool for Universal Planning on PDDL+ Problems”. In: *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling*. 2009 (cit. on p. 27).

[Pri] PricewaterhouseCoopers (PwC), Strategy (part of PwC network) and the Research Centre for Energy and Sustainable Development of the Macedonian Academy of Sciences and Arts (MANU) in North Macedonia. *The Strategy for Energy Development of the Republic of North Macedonia until 2040*. URL: https://economy.gov.mk/Upload/Documents/Adopted%20Energy%20Development%20Strategy_EN.pdf (cit. on pp. 32, 34).

[PWF+18] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, E. Ott. “Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* (2018). DOI: https://doi.org/10.1063/1.5028373 (cit. on p. 42).

[Ras] Raspisaniye Pogodi Ltd. *Historical Weather Data*. URL: https://rp5.ru/Weather_in_the_world (cit. on p. 50).

[RNHB17] M. Q. Raza, M. Nadarajah, D. Q. Hung, Z. Baharudin. “An intelligent hybrid short-term load forecasting model for smart power grids”. In: *Sustainable Cities and Society* (2017). DOI: https://doi.org/10.1016/j.scs.2016.12.006 (cit. on pp. 37, 47, 49).

[SGZG10] A. S. Sambo, B. Garba, I. H. Zarma, M. M. Gaji. “Electricity Generation and the Present Challenges in the Nigerian Power Sector”. In: *In Proceedings of the Conference: 21st World Energy Congress (WEC) 2010*. 2010 (cit. on p. 34).

[Sha21] A. Shaikh. *Energy Optimization in HVAC Using Automated Temporal Planning*. 2021 (cit. on pp. 43, 44).

[SK16] H. Son, C. Kim. “Short-term forecasting of electricity demand for the residential sector using weather and social variables”. In: *Resources, Conservation and Recycling* (2016). DOI: https://doi.org/10.1016/j.resconrec.2016.01.016 (cit. on pp. 36, 47, 49, 51).

[SK20] H. Son, C. Kim. “A Deep Learning Approach to Forecasting Monthly Demand for Residential–Sector Electricity”. In: *Sustainability* (2020). DOI: https://doi.org/10.3390/su12083103 (cit. on pp. 33, 34, 36, 49).

[Stra] Struja(dot)mk. *Interruptions in Electricity Supply for Wednesday (18.10.2023)*. URL: https://struja.mk/prekini-vo-snabduvanje-na-elektrichna-energija-za-sreda-18-10-2023/ (cit. on p. 32).

[Strb] Struja(dot)mk, owned by Energy Delivery Solutions (EDS). *Electricity Related News in North Macedonia*. URL: https://struja.mk/za-nas/ (cit. on p. 32).

[Str08] G. Strbac. "Demand side management: Benefits and challenges". In: *Energy Policy* (2008). DOI: https://doi.org/10.1016/j.enpol.2008.09.030 (cit. on pp. 21, 23).

[Sub] A. K. R. Subrata Mukhopadhyay. "Demand side management and load control — An Indian experience". In: *Proceedings of the IEEE PES General Meeting* (). DOI: https://ieeexplore.ieee.org/document/5589589 (cit. on p. 33).

[Tho] Thomson Reuters. *Electricity Regulation in South Africa*. URL: https://uk.practicallaw.thomsonreuters.com/w-0185347?contextData=(sc.Default)&transitionType=Dfault&firstPage=true#co_anchor_a502112 (cit. on p. 33).

[UMA+22] R. Usman, P. Mirzania, S. W. Alnaser, P. Hart, C. Long. "Systematic Review of Demand-Side Management Strategies in Power Systems of Developed and Developing Countries". In: *Energies* (2022). DOI: https://doi.org/10.3390/en15217858 (cit. on pp. 21–24, 33, 91).

[Uni] University of California, Irvine. *Unplanned Outages*. URL: https://www.fm.uci.edu/programs/unplanned-outages.php (cit. on pp. 17, 34).

[VAKP20] G. Veljanovski, M. Atanasovski, M. Kostov, P. Popovski. "Application of Neural Networks for Short Term Load Forecasting in Power System of North Macedonia". In: *International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)* (2020). DOI: https://doi.org/10.1109/ICEST49890.2020.9232674 (cit. on pp. 37, 49).

[Wor21] Worldometers. *South Africa Population*. 2021. URL: https://www.worldometers.info/world-population/south-africa-population/ (cit. on p. 33).

[WWT+19] S. Wen, Y. Wang, Y. Tang, Y. Xu, P. Li, T. Zhao. "Real-Time Identification of Power Fluctuations Based on LSTM Recurrent Neural Network: A Case Study on Singapore Power System". In: *IEEE Transactions on Industrial Informatics* (2019). DOI: https://doi.org/10.1109/TII.2019.2910416 (cit. on pp. 18, 39, 40, 49, 91).

[XWL+16] C. Xu, G. Wang, X. Liu, D. Guo, T.-Y. Liu. "Health Status Assessment and Failure Prediction for Hard Drives with Recurrent Neural Networks". In: *IEEE Transactions on Computers* (2016). DOI: https://doi.org/10.1109/TC.2016.2538237 (cit. on p. 39).

[YWL16] M. Yuan, Y. Wu, L. Lin. "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network". In: *IEEE International Conference on Aircraft Utility Systems (AUS)* (2016). DOI: http://dx.doi.org/10.1109/AUS.2016.7748035 (cit. on p. 18).

[ZWLB17] S. ZHANG, Y. WANG, M. LIU, Z. BAO. "Data-Based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM". In: *IEEE Access* (2017). DOI: https://doi.org/10.1109/ACCESS.2017.2785763 (cit. on pp. 17, 18, 31, 34, 38–41, 47, 49, 50, 53).

All links were last followed on October 27, 2023.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Stuttgart, 28.10.2023

place, date, signature