University of Cincinnati

Date: 3/12/2021

I, Nicholas Maltbie, hereby submit this original work as part of the requirements for the degree of Master of Science in Computer Science.

It is entitled:

Integrating Explainability in Deep Learning Application Development: A Categorization and Case Study

Student's name: **Nicholas Maltbie**

This work and its defense approved by:

Committee chair: Nan Niu, Ph.D.

Committee member: Raj Bhatnagar, Ph.D.

Committee member: Anca Ralescu, Ph.D.

38173

Integrating Explainability in Deep Learning Application Development

A Categorization and Case Study

A thesis submitted to the

Graduate School

of the University of Cincinnati

in partial fulfillment of

the requirements for the degree of

Master of Science

in the Department of Electrical Engineering and Computer Science

of the College of Engineering and Applied Science

by

Nicholas Maltbie

B.S. University of Cincinnati

May 2021

Committee

Professor Nan Niu Ph.D. (Committee Chair)

Professor Raj Bhatnagar Ph.D.

Professor Anca Ralescu Ph.D.

Integrating Explainability in Deep Learning Application Development

A Categorization and Case Study

Nicholas Maltbie

Abstract

Artificial intelligence and machine learning (ML) solutions are being integrated into more software solutions than ever before. The inner working of ML models is difficult to understand making them black boxes. Explainable artificial intelligence (XAI) tools have been developed to make these ML models more understandable. In this work, we reviewed existing XAI solutions and their implications. We have grouped these tools based on how they operate and what kinds of information they can provide to the stakeholders. To illustrate this, we conducted a case study with the Metropolitan Sewer District of Greater Cincinnati to evaluate the use of XAI to make ML solutions more accountable and understandable. As part of our case study methodology, we investigated how these XAI tools can be integrated into the ML development pipeline. There is a clear need for explainability in ML based software and our case study and research investigated how these XAI tools can enhance ML testing and validation.

Contents

List of Figures and Tables

Acknowledgements

I want to thank my committee chair and advisor, Dr. Nan Niu, for supporting my academic achievement and research. Additionally, I want to thank my committee members, Dr. Anca Ralesu and Dr. Raj Bhatnagar, for their support and advice for this thesis.

I want to thank Reese Johnson and Matthew VanDoren from the Metropolitan Sewer District of Grater Cincinnati for their support, feedback, and resources to complete this research and case study.

I want to thank my mom, Dr. Cathy Maltbie, for help proofreading and editing this thesis, my friends, Nicholas Driggs, Richard Chiang, and Andrew Aldrich.

Chapter 1

Introduction

Artificial Intelligence (AI) has become integrated into decisions and shapes our daily life, via news feeds, search results, and even shopping recommendations. To understand the scale of AI, the International Data Corporation (IDC) forecasts that global investment in AI will grow from \$50.1 billion in 2020 to more than \$110 billion by 2024 [29]. According to a report from Gartner, the value of AI-derived business is expected to grow from \$692 billion in 2017 to \$3.92 trillion in 2022 [35].

This unstoppable growth of AI has also reached into the public sector. For example, the European Commission envisions that AI could be used to serve citizens 24/7 in more accessible ways [25]. However, some of these implemented public AI services have had unintended, and sometimes harmful, consequences. In the U.S., for instance, AI was used to allocate caregiver hours for people with disabilities, but reduced the hours for caregivers in hundreds of cases without any explanation, leaving them without the ability to contest the decision made by these proprietary algorithms [59].

As with any new technology, mistakes while using AI are inevitable, but the lack of *explainability* can be concerning for both citizens and public organizations. AI-based decision processes lack transparency, making it difficulty to verify accountability, fairness and responsibility of these systems. Explainable Artificial Intelligence (XAI) attempts to address these concerns by making AI more understandable to users, increasing the trust and reliability of such AI-based systems.

Given the importance of explainability, many XAI tools are used to generate explanations from AI systems. Deep learning is a prominent type of AI-based system that relies on neural networks which emulate the structures of human brains and learning. In recent advancements, these deep learning models have been able to achieve near-human accuracy in various types of classification and prediction tasks including images, text, speech, and video data [9]. These deep learning models are opaque and are sometimes referred to as "black boxes" due to their closed off nature and difficulty in understanding how they operate [23]. One influential XAI tool is LIME (Local Interpretable Model-agnostic Explanations) proposed by Ribero *et al.* [47]. LIME can approximate a black box model by sampling the local neighborhood of any prediction. An illustrative example given by Ribero *et al.* [47] is that once a model predicts a patient is sick, LIME shows that the "sneeze" and "headache" symptoms contribute to the prediction whereas "no fatigue" is evidence against. XAI tools such as LIME that identify features that contribute to a prediction are just one type of explanation. Other tools may extract rules, visualize salience maps, or use other methodologies [1].

There has been significant innovation in and development of new XAI tools in the past few years. A 2018 survey from Guidotti *et al.* [23] represents an overview of XAI tools for many types of machine learning. However, since the publication of that survey, there have been many XAI tools specifically created to better understand deep learning based models. The rate of innovation for XAI tools for deep learning specifically has led to many new tools being created, such as SHAP [47], DeepLIFT [50], RuleMatrix [40] and more. These XAI tools aim to provide transparency and build trust in AI-based software systems. The current body of research lacks an overview of these recently developed tools and how to integrate them into the software development process.

Miller *et al.* [39] argue that most XAI researchers are currently building tools for themselves rather than for the intended user of the product. Even seminal work on LIME scored the lowest on Miller *et al.*'s "data-driven" criteria, because Riberio *et al.* [47] constructed their own understanding of how people might evaluate explanations and recruited human subjects from Amazon Mechanical Turk to perform the experiments. Behavioral experiments conducted with lay persons are simplification and therefore cannot replace putting an explanation into a real-world application and letting an actual end user (typically a domain expert) test it [16].

To gain insights into application-grounded XAI evaluation, I conducted a case study with the Metropolitan Sewer District of Greater Cincinnati (MSD) on how public services might exploit deep learning to predict combined sewer overflows (CSOs). Combined sewer systems transport various sources of water from residential, industrial, and commercial customers as well as storm runoff. A crucial problem is handling CSO events when the system is overwhelmed by surges of water and the combined sewer system is forced to discharge untreated water into the local environment. MSD has developed a sensor network collecting real-time water flow data along with the availability of contributing sources like the rainfall data. They have a keen interest in deep learning techniques and utilizing their high predictive accuracy as well as explainability.

In order to best utilize current XAI tools for this case study, I conducted a softgoal oriented review of applicable XAI tools. Softgoals represent goals related to quality attributes such as reliability, explainability, and robustness and are critical for success but are difficult to quantifiably measure and do not have clear cut achievement criteria [22]. Each of these tools provides and creates explanations from a deep learning model in a different manner. Understanding these differences and how they relate to these softgoals of explainability is important to best understand how they can provide explanations to end users. Gilpin *et al.* [17] have discussed how an explanation must be fit to a stakeholder's background. Understanding how each tool operates and the kinds of explanations they provide are vital for providing applicable explanations to stakeholders.

In this thesis, I make three main contributions: a softgoal-oriented categorization and organization of existing XAI tools; a goal-oriented question-metric analysis to quantitatively measure three state-of-the-art XAI tools as part of a case study; and an analysis applying these XAI tools to the machine learning engineering development process. In this thesis, I review existing work and background in Chapter 2, define softgoals for explainability and categorize existing XAI tools in Chapter 3, describe the background, methodology, and results of the case study in Chapter 4, discuss the implications of this work in Chapter 5, and state concluding remarks and recommendations for future work in Chapter 6.

Chapter 2

Literature Review

Before we can establish the need for explainability applications, we first build a basic understanding of deep learning and XAI. There have been many new XAI tools published and improved in the past few years, but there have only been a few surveys of XAI tools within that same time frame. Therefore, there are many tools and methodologies that are not well reviewed in existing research. Current surveys, including Guidotti *et al.* from 2018 [23], Gilpin *et al.* from 2018 [17], Adadi *et al.* from 2018 [1], and Arrieta *et al.* from 2020 [3], are vital to understand and frame XAI tools and I seek to build upon these works. Arrieta *et al.* from 2020 [3] is particularly relevant as it surveys some of the most recently published XAI tools and research.

For the purpose of this review, I focus on an overview of XAI tools related to deep learning and recent research into the ML engineering development process. Through related research, I define the necessary background needed when overviewing existing research. Given the vast base of existing surveys and research of XAI in general, I have limited my work to mainly focus on deep learning.

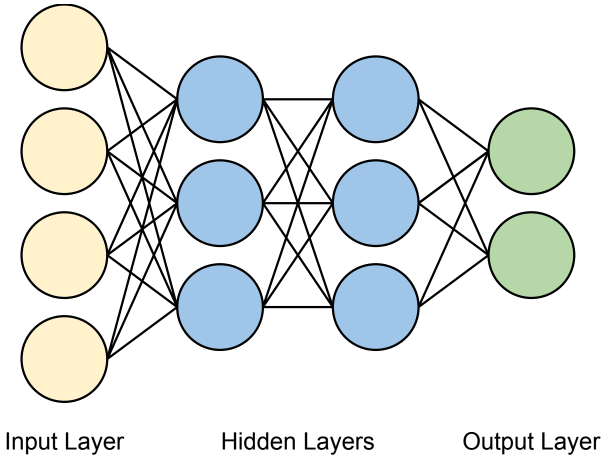


Figure 2.1: Basic structure of a deep learning model. Inputs are given to the model through the input layer and propagated through the model by using a weights matrix between each layer representing connections between nodes. The final output layer represents the model's prediction.

2.1 Deep Learning

My work focuses on deep learning and applications of XAI to deep learning software solutions. Deep learning has become a hype word used in marketing, research, and related fields. There can be many definitions and interpretations for the term deep learning but they all generally define ML-based structures modeled from Artificial Neural Networks (ANN) [64]. Machine learning (ML) is a broader term that includes many more technologies such as clustering, pattern mining, variance analysis, and other applications. For the purpose of this work, deep learning is defined an ANN-based ML which contains a similar structure to Figure 2.1 with nodes organized into layers with an input layer, hidden layers, and an output layer. Deep learning models are usually trained in a supervised manner; they are initialized with random weights and then shown labeled training samples to use backpropagation to compute improvements to the deep learning model.

I am specifically focusing the scope of this study on deep learning given both its prominence in the past few years and its inherently opaque nature. These deep learning models are able to utilize the massive amounts of data collected over recent years. These deep learning solutions range over many applications including facial recognition [49], cancer diagnosis

[28], and autonomous driving [44]. Despite this success, deep learning models are also black boxes and may have biases, errors, or various flaws that are nearly impossible for humans to easily identify [23]. Understanding the components of these deep learning models and how they operate are vital to understanding how they can be explained.

2.2 ML Testing and Validation

Deep learning models represent a large investment for stakeholders due to the cost of data and resources used in creating models. There is a need in current research to ensure that ML solutions perform as desired. A summary of the current state of the art technology for ML testing and validation has been completed by Zhang *et al.* [63]. According to Zhang *et al.*, ML testing actively seeks to detect differences between the actual and desired behaviour of ML systems [63]. Large technology companies that use ML (e.g., Google, Amazon, and Microsoft) have all published research findings on the software engineering process and specifically include testing and validation of models. The performance of these tools is vital to establishing business goals and improving products for stakeholders. For example, Polyzotis *et al.* in 2019, from Google, found that correcting a bias in the ML system for recommending apps through the Google Play Store increased install the rate of apps by 2% [43].

Although this thesis specifically focuses on deep learning, this kind of testing and validation can be used for any type of ML process and much of the published work emphasizes its importance. The terminology for the next section will focus on broader ML instead of more specific deep learning.

2.2.1 Data Validation

Currently, much of the research in the field focuses on data validation and cleaning as the primary use of ML testing and validation. My work seeks to add explainability achieved through XAI tools. As of right now, data validation and cleaning are the main components of ML testing and validation and there is not a strong focus on XAI tools. As noted by work such as Schelter *et al.*, the quality and quantity of data are vital to developing ML models [48]. Research by Polyzotis *et al.* notes that data errors degrade the quality of models which can then create a negative feedback loop when these models and results are used to train future models [43]. Due to the great importance of data in ML, this topic is vitally important and will be discussed throughout this thesis. It is imperative that large development teams have a process for preparing data and effectively utilizing these data in their ML models. Work such as Amershi *et al.* describes the significance that data play in model training and validation and outlines an ML workflow to guide the development of future projects at Microsoft [2]. Breck *et al.* analyzed the performance of Google teams that deployed ML-based software [7]. Polyzotis *et al.* published findings with Google to present a data validation system for ML software development [43].

These kinds of processes and data validation help identify and resolve data errors. These errors can range from simple problems in storage to more complex code and processing in highly interconnected systems [43]. Understanding the data for a problem is critical to any ML process and any errors in the data must be handled or acknowledged.

2.2.2 ML Engineering Workflow

Data preparation is essential in the ML engineering workflow for developing software. Examples of this include the ML workflows proposed by Amershi *et al.* [2] and Polyzotis *et al.* [43]. In this thesis, I will discuss how explainability and XAI tools can be integrated into the ML development workflow. Following a standard workflow can help organize larger teams and provide a framework for developing new software. Workflows help facilitate the complexities in ML and the accelerating rate at which it is being integrated into workplaces. A simplified form of the ML workflow proposed by Amershi *et al.* [2] is shown in Figure 2.2. In the original paper, this process has 9 stages of development: this simplified view is used to show the major phases of the workflow as they influenced my research. These phases are

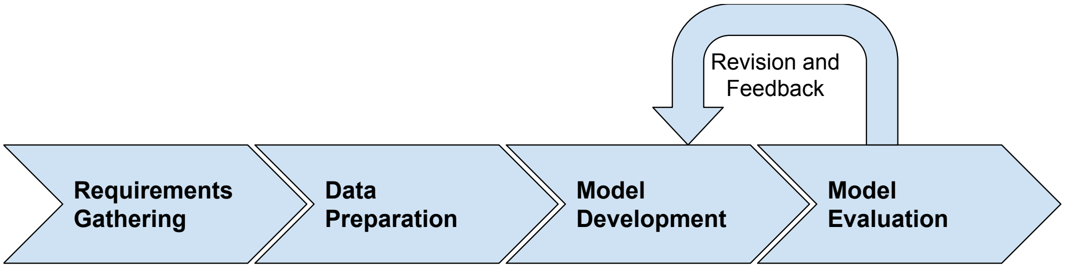


Figure 2.2: Simplified diagram of the ML workflow proposed by [2] with four major phases (1) Requirements Gathering, (2) Data Preparation, (3) Model Development, and (4) Model Evaluation.

similar to a waterfall-like software development workflow where a single stage of development is completed, then the next stage of development is started. There is a looping flow between model evaluation and model development since results of evaluating the model may require changes to the model and design of the solution.

Software testing is a well-established development practice and there are several key differences between software and ML testing. According to Zhang *et al.*, while software testing only tests code, ML testing will verify code and data used in the ML system [63]. There are many different ways to integrate ML testing into workflows and Zhang *et al.* goes into detail in their survey about these methodologies and their uses. The importance of data and code testing is reflected in the various workflows and solutions proposed by methodologies from Amershi *et al.* [2] and Breck *et al.* [7]. Verification of data and code are vital to ensuring ML systems will perform as desired.

2.2.3 ML Testing Softgoals

ML testing softgoals represent the desired behaviour of the ML systems. Establishing various goals of ML testing forms the foundation for this thesis. I will expand upon the goal of interpretability and explainability and how these XAI tools can help satisfy this softgoal. These desired goals and methodologies may differ depending on a project but some key ideas remain prominent across all types of work. The terms defined below are derived from a survey by Zhang *et al.* and their summary of ML testing properties [63] as well as various other recently published works.

- **Data Validity** concerns understanding the data for a problem and ensuring its correctness. This is a well studied component from work such as Breck *et al.* studying teams at Google [7]; Zhang *et al.* notes the importance of identifying data bugs as a part of correctness [63]. There is a great deal of specialized research into how to verify data validity such as Polyzotis *et al.* [42] and their work analyzing challenges of organizing data in large interconnected systems. Open-source tools have been developed to specifically address data validation such as Tensorflow Data Validation from Caveness *et al.* [8].

- **Model Performance** concerns verifying the appropriateness of an ML model and how well it achieves the desired behavior. This includes assessing model correctness through measures such as accuracy, precision, recall and others relevant to a problem [63]. This also may include measures of how much performance varies between training and testing [11], and also ensuring that a model is relevant and not over or under fit for a given problem.

- **Robustness** concerns a non-functional characteristic pertaining to an ML system's fault resistance. Zhang *et al.* describe this as the ability to work with invalid or stressful conditions and can be tested through various means [63]. Robustness also includes the system's ability to recover from failures and the importance of results provided to stakeholders as described by Breck *et al.* [7].

- **Interpretability** concerns how well a human is able to understand and evaluate an ML system. This can be analyzed through various means such as a human assessment of models described by Zhang *et al.* [63]. Other methodologies for ML development such as Amershi *et al.* express the need for interpretability when debugging as an important non-functional requirement [2]. Breck *et al.* further expresses the need to understand how these ML models make decisions to debug them [7].

Interpretability is important to ML testing as it can lead to finding issues or detecting biases in a model. For example, Lapuschkin *et al.* were able to identify a strong relationship between the ImageNet [14] class of horses to a copyright symbol in the image [33]. Being able to identify the cause of these errors within the black box structure of the model and understand how the model works has been shown to be vital to debugging and improving the performance of the models.

2.3 XAI Tools Overview

Explainability and explainable artificial intelligence (XAI) are critical to providing insights into how these deep learning systems operate. As discussed earlier, this could involve finding a correlation between image identification and a copyright symbol [33] or identifying bias in a dataset to improve user interaction with the system [43]. Errors such as these may be acceptable when they occur in a negligible percentage of cases for showing an advertisement or differentiating animals in images but this is not acceptable when the decisions may have safety implications or are critical topics.

In AI, the high level of difficulty for a system to provide a suitable explanation for how it arrived at an answer is referred to as the black box problem [1]. This difficulty is particularly prominent for deep learning models, because an ANN trained end-to-end can be as complex as an accurate explanation of why the model works [18]. The complexity can be illustrated by ResNet [24], which incorporates about 5×10^7 learned parameters and executes about 10^{10} floating point operations to classify a single image. XAI tries to demystify the black boxes as they begin making decisions previously entrusted to humans. Thus, explainability—the ability to interpret the inner workings or the logic of reasoning behind the decision making—helps an AI system to achieve the following:

- **accountability**: justifying decisions and actions,
- **fairness**: having impartial treatment and behavior,

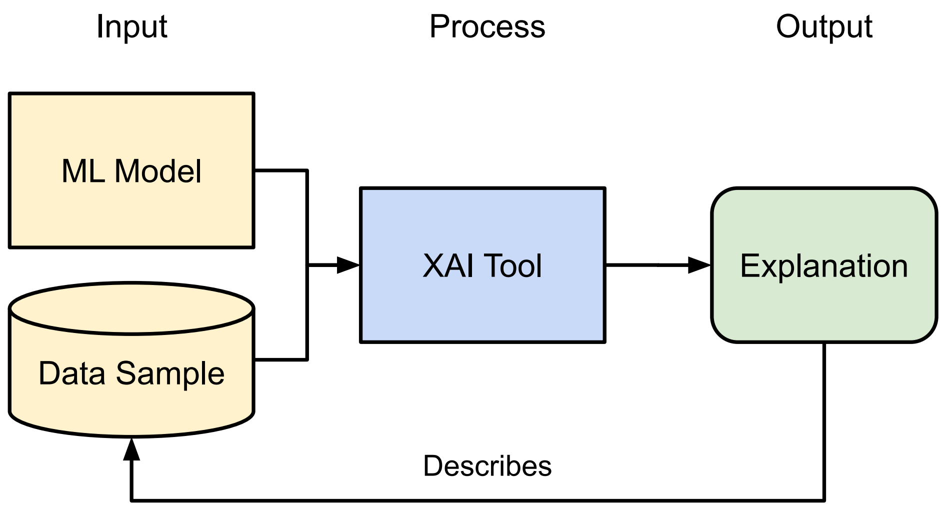


Figure 2.3: Process of generating an explanation for a data sample given an XAI tool. This high-level process describes the general methodology that almost every XAI tool follows.

- **responsibility**: answering for one's decisions and identifying errors or unexpected results.
- **transparency**: describing and inspecting the mechanisms through which decisions are made.

XAI tools generate explanations from an ML model and a data sample with the generated explanation describeing that sample. Figure 2.3 shows this process visually. Adadi and Berrada [1] identified 17 XAI techniques by surveying 381 papers published between 2004 and 2018. According to the survey, most recent work done in the XAI field offers a *post-hoc*, *local* explanation. Because only a few models, such as linear regression or decision trees, are inherently interpretable, generating *post-hoc* explanations is necessary for complex models like DNNs. *Post-hoc* XAI tool can therefore be applied to any classifier or regressor that is appropriate for the application domain—even those that are yet to be proposed [47].

2.3.1 Classifying XAI Tools

Understanding how XAI tools operate similarly is important to evaluating how they can be used and how they can help satisfy non-functional requirements. In this thesis, I focus on investigating how a set of XAI tools can be used to satisfy various aspects of explainability. What XAI tools do can be classified by how they emulate processing data to draw connections between inputs and outputs. Gilpin *et al.*'s taxonomy [17] shows that XAI processing can (1) extract rules to summarize decisions, (2) create a salience map to highlight a small portion of the computation which is most relevant, and (3) employ a simplified proxy that behaves similarly to the original model. For instance, Benítez *et al.* [6] transformed DNNs to fuzzy rules through an equivalence-by-approximation process, Simonyan *et al.* [51] produced a salience map by directly computing the input gradient, and Ribeiro *et al.* [47] used a local linear model in LIME as a simplified proxy for the full model.

While proposing DeepLIFT [50], Shrikumar *et al.* examined another set of XAI tools by how they provide explanations. Since each XAI tool operates slightly differently there is a hierarchy between how similar different methodologies are, with one XAI tool potentially using multiple methodologies.

- **Proxy Model** involves generating a simplified model similar to the rule extraction defined by Gilpin *et al.*'s taxonomy [17]. This can be further subdivided into *Global Approximation* or *Local Approximation.* Global approximations involve approximating the model globally and how it will behave for every sample. Global approximations include simplifying a model to decision trees such as DeepRED from Zilke *et al.* [65] or Benítez *et al.* [6] transforming models to fuzzy rules. Local Approximations create a local approximation of a model for each given sample. This includes methodologies such as LIME from Ribeiro *et al.* [47] which uses a linear approximation for each sample locally. SHAP from Lundberg *et al.* [36] also uses local approximations through shapely values.

- **Backpropagation** uses the reverse of forward propagation when specifically explaining deep learning structure models. This allows an XAI tool to look at a prediction and move backwards through a model to explain where the influence came from. Two popular methods for this are *Gradient-Based* and *Shapely-Based. Gradient-based* methodologies use the proportional weights within the deep learning model to propagate influence through an ANN structure such as Laywerwise-Relevance Propagation proposed by Bach *et al.* [4]. *Shapely-based* backpropagation uses shapely values which measure the average effect of an input over all possible outputs based on a "game theory"-like approach [50]. XAI tools such as DeepLIFT proposed by Shrikumar *et al.* [50] and SHAP proposed by Lundberg *et al.* [36] use backpropagation.

- **Permutation** involves occluding or changing input space to find the most relevant inputs. This methodology takes quite a bit of computational time and is useful for testing and verification for fuzz-testing-like methodologies. Zeiler and Fergus proposed a methodology for achieving interpretability through occluding sections of input [61]. This methodology is quite expensive and does not scale for larger datasets.

2.3.2 Limitations of XAI Tools

With the increased usage of XAI techniques, evaluating their efficacy becomes important to inform practitioners about effective tool adoption. Miller *et al.*'s survey of 23 XAI papers [39] found that rigorous human behavioral experiments are not currently being undertaken. As the verb to explain is a three-place predicate, "Someone explains something to someone [26]", Miller *et al.* [39] argue that most XAI tools explain things (e.g., feature or neuron importance) to the AI researchers but *not* to the intended users. Doshi-Velez and Kim [16] further argue that the best way to show how an XAI technique works is to evaluate the tool by consulting domain experts grounded in the exact application task. Although costly, the application-grounded evaluations provide direct and strong evidence (or lack thereof) of XAI's fulfillment of the requirements.

2.4 Explainability as a Non-Functional Requirement

For this thesis, I focus on Explainability as a non-functional requirement and how it can be integrated into the ML development process. In software engineering, *functional requirements* describe what the system does, whereas *non-functional requirements* (NFRs) focus on how well the system does it [12]. Making classifications, recommendations, and predictions are among the common functional requirements of an AI system, and doing so in an explainable way is often regarded as a non-functional concern [32]. Therefore, researchers consider explainability to be an NFR.

In a survey study with 107 participants (90 from Brazil and 17 from Germany), Chazette and Schneider [10] gathered participant expectations for an explanation. Chazette and Schneider's online questionnaire used a hypothetical scenario where participants would use a vehicle's AI-based navigation system while driving on a route they had traveled before; however, the AI suggested a different route than usual. Of the 103 codes analyzed from all the responses, 36 (35%) expressed desire in knowing *what* specific piece of information supported and influenced the suggestion, 12 (12%) wanted to know the *how* of the algorithm's inner reasoning, and 55 (53%) expressed willingness to understand *why* something happened (e.g., "why the [usual] route is not being suggested" and "benefits of the new route when compared to the usual" [10]).

The survey results clearly show that people have different requirements of explainability. Chazette and Schneider [10] pointed out that eliciting explainability should also consider laws and norms, cultural and corporate values, domain aspects, and practical project constraints such as time and budget. The European Union, for instance, debated a general "Right to Explanation" [19] which is partly enshrined in certain regulations [45]. Such policies, along with globally emerging ethics guidelines [30], are making AI—especially AI in citizen services—more auditable.

NFRs may interact; the attempts to achieve one NFR can hurt or help the achievement of another. For example, generating a *post-hoc* explanation imposes additional computational overhead, possibly hurting an AI service's responsiveness. Paradoxically, 35% of the codes in [10] corresponded to responses in which users perceived explanations as a way to reduce obscurity due to having more information about an AI system and its outcomes, and yet 15% cautioned that too technical or lengthy explanations might add more obscurity. Recognizing the trade-offs between explainability and other NFRs is therefore important for prioritizing requirements and making design choices.

In summary, the requirements engineering literature suggests that explainability is an NFR, or a softgoal whose satisfaction is a matter of degree without a clear-cut criterion [12]. Understanding how, and to what degree, XAI tools satisfy the explainability softgoal is the focus of this thesis. I will use both a review of existing literature as well as a data-driven case study to investigate an application of XAI tools.

Chapter 3

Requirements Engineering of XAI Tools

There has been a recent acceleration in the development of explainability and deep learning tools and methodologies. This large body of work, identifying which tool is best fit for a specific task, may be difficult especially given the rapid pace of innovation in deep learning methodologies. Categorization of tools and explainability do exist. Adadi and Berrada [1] categorize XAI tools, and recent surveys including work from Gudotti *et al* [23] and Arrieta *et al* [3] provide great overviews of existing XAI tools. However, these surveys and categorizations do not provide a unified approach for evaluating tool explainability as an NFR. In order to evaluate an XAI tool for a given task, I am proposing a set of softgoals for assessing the fitness for purpose of XAI tools from reviewing existing requirements engineering and explainability literature.

3.1 Explainability Softgoals

As defined earlier, NFR softgoals represent quality attributes such as reliability, explainability, and robustness that are critical to system operation but do not have clear cut criteria of completion [22]. These are in opposition to functional requirement goals which are critical to success and have clearly defined criteria for completion. These functional requirements may include actions or capabilities of a system such as providing information to sending information to a server. Non-functional requirement softgoals represent how these goals can be achieved, such as providing information in an accessible manner or a level of robustness or fault tolerance when communicating with a server.

Explainability is considered an NFR for ML-based predictions. XAI tools attempt to satisfy this explainability requirement through different methodologies. Gilpin *et al.* define a taxonomy for grouping types of explanations [17]. The work of Köhl *et al.* defines explainability as an NFR and proposes a methodology for evaluating the quality of explanations [32]. As noted by Gilpin *et al.* [17] and Köhl *et al.* [32], the terminology associated with XAI is often confusing and sometimes contradictory. To address this lack of unified terminology and evaluation of explainability, I present five softgoals for evaluating explainability of an ML system. These five softgoals represent aspects or abilities of a XAI tool and deep learning model.

1. *Trust-ability* - Stakeholders' belief in a given model or prediction.
    (a) *Persuasive* - How well can we convince a user to trust that a prediction is correct?
    (b) *Transparency* - How well can we trust and understand the workings of a model including its flaws?
2. *Interpret-ability* - Stakeholders' understanding of a given model.
    (a) *Training* - How did a model get here, from data, configuration, and parameters?
    (b) *Composition* - What is the model composed of and how do these parts interact?
3. *Justify-ability* - The tool's ability to explain an individual prediction to stakeholders.
    (a) *Comprehensibility* - How well can a given stakeholder understand this explanation?

   (b) *Completeness* - How true is this explanation to the original model?

4. *Apply-ability* - This tool's ability to be applied to new data or situations.

   (a) *Performance* - Can we trust this model to perform well in an unbiased manner with a new dataset?

   (b) *Boundaries* - When do we expect this model to break, and will these edge cases be handled gracefully?

5. *Disrupt-ability* - Level of new insights provided by an ML model.

   (a) *Contrast* - How much does this contradict or support existing beliefs?

   (b) *Creativity* - Does this solution provide a new perspective to the issue or problem?

Each of these softgoals is derived from existing literature and requirements of ML models. Through this case study and from reviewing work such as Breck *et al.* [7], I have found that software engineers want to know why an ML system may perform better and how they can take these insights to better understand how these characteristics can be integrated into future work. *Trust-ability* is a major focus of explainability of XAI tools, since establishing trust in a model is essential, as seen with LIME [47]. When XAI tools provide explanations, they are either trying to *justify* an individual prediction or are attempting to make the model more *interpret-able* or transparent. The softgoal of *apply-ability* focuses on under what conditions one expects a model to work. This is important in traditional software development just as with ML software development. A system should react in a predictable way and perform as expected even when using new data or new situations. These five abilities are not an exhaustive list of softgoals, as some stakeholders may require reliability, robustness, security, privacy, or other softgoals as listed in existing literature from Zhang *et al.* [63]. These five softgoals represent the scope and definition of explainability used in this thesis.

While not discussed in this thesis, these components of explainability can be satisfied by other means than XAI tools. Work such as Breck *et al.* [7] established methodologies for evaluating *trust-ability* in an ML system through automated testing and data validation [7]. Baylor *et al.* [5] proposed an automated tool for data cleaning, validation and testing to improve model performance. Automated testing and data validation help establish explainability by helping developers debug and understand their ML-based systems more. Work such as Baylor *et al.* [5] and Polyzotis *et al.* [43] emphasizes the importance of data validation to identify errors and fix broken parts of an ML systems.

3.2 Significance of Softgoals

Understanding how explainability is useful to non-ML systems is critical to frame how explainability can help ML systems. An analogy of a hospital's emergency room and admitting patients is useful to understand why explainability is critical in any process. In this analogy, a patient comes into the hospital and can either be taken for immediate care or sent to a waiting room. The staff will observe the patient's symptoms and identify the best treatment option. As the hospital staff make decisions, they are following a set of rules and intuition from previous experiences. This established system of rules helps the hospital staff make consistent and correct decisions. As part of the procedure, it is common for a hospital to log patient symptoms and treatment while they are at the hospital.

In this analogy, a false positive event would be sending a patient to immediate care when they could have spent time in the waiting room. A false negative would be sending a patient to the waiting room when they needed immediate care. Making an incorrect prediction will use extra resources and possibly lead to a worse outcome for the patient. Each of our softgoals from explainability can be framed with the *procedure* for admitting a patient to the waiting room or for immediate care.

Table 3.1 illustrates how each of the softgoals is utilized to evaluate explainability in the

Table 3.1: Description of how each of the softgoals relates to the healthcare analogy for how explainability can be used in any procedure or system.

Softgoal	**Healthcare Example**
(1) Trust-ability	Hospital staff trust the procedure to help patients.
(2) Interpret-ability	Hospital staff interpret how the procedure will provide care to different patients.
(3) Justify-ability	The procedure logs and justifies why a patient was given a specific treatment by hospital staff.
(4) Apply-ability	The procedure applies correct decisions for patients with new symptoms or hospitals with new staff.
(5) Disrupt-ability	The new procedure disrupts and improves patient treatment to hospital staff experience.

patient treatment procedure. Let us assume for the healthcare procedure the hospital uses a rule-based system and explainability would naturally be part of the system and considered in all steps of the process. The staff or patient can look at the list of rules and then know exactly why treatment is recommended. For example, there could be a rule that if the patient has a fever and cough they need to be taken in for immediate treatment. The hospital may need this explainability for accountability or legal requirements, but explainability can also help improve the system and give those following the system more confidence in the validity of the procedure. This confidence and explainability is easy to establish with an open decision process such as a set of rules. However, if the system uses an opaque, black box solution such as deep learning, the whole process must include explainability in order to provide confidence and accountability to the system. Imagine telling a patient that they need to go to the waiting room for three hours because a proprietary algorithm predicted this as the best outcome. This could be seen as unreasonable and not create a sense of confidence in the trustworthiness of the system. While there is a need for explainability even when systems are not using complex deep learning algorithms, when they use deep learning it is even more important to stakeholders.

3.3 Methodology of Review

These softgoals provide necessary components of explainability to stakeholders. To illustrate the significance of these softgoals, I surveyed existing XAI tools based on how well they satisfy these softgoals. It is important to be able to both quantitatively and qualitatively evaluate each of the XAI tools based on how well they satisfy explainability. As discussed in Section 2.3, there are many existing XAI tools and I reviewed a range of tools that provide different kinds of explainability as well as different methodologies for explaining the deep learning model. The well-defined XAI tools used for this thesis are listed below.

- **RuleMatrix** [40] provides a global approximation and proxy model of a black box solution. This is achieved by converting a deep learning model into a set of rules.
- **LIME** [47] uses a local approximation and a linear gradient through a proxy model to explain black box solutions. It can identify the most significant features for a given prediction and which class those features support.
- **Layer-wise Relevance Propagation** (LRP) [4] uses a local approximation achieved through backpropagation and provides a saliency map. Similar to LIME, this will identify how much each feature contributes to a given class.
- **DeepLIFT** [50] also provides a local approximation and saliency map. This uses shapely values and backpropagation as opposed to the proportional attribute used in LRP.
- **TFX** [5] is not explicitly an XAI tool but provides automated data validation and cleaning. Data are a major component of explainability in deep learning are data so evaluating this tool provides an interesting comparison to other XAI tools.

In order to evaluate how well each of these tools satisfies the softgoals of explainability, I reviewed the original proposal of the tool and evaluated how well each satisfies the softgoals.

Each of these tools has had many variations, improvements, and adjustments published that may help to satisfy a specific softgoal. I will limit my scope to evaluating how the tool was originally proposed in an “out of the box manner” and not how it has been improved and expanded. This analysis can be repeated for any XAI tool as this process can be used as a framework to evaluate how well a tool satisfies explainability. This analysis is similar to the XAI tool taxonomy proposed by Gilpin *et al.* [17].

3.3.1 Explainability Questions

In Section 3.1, I proposed softgoals for evaluating explainability. Simply applying these softgoals to an XAI tool is difficult as they cannot easily be measured. In order to evaluate these tools, I have developed a set of questions that can be asked when attempting to use a tool to evaluate an ML-based system and its development. The general stages of the ML engineering workflow were discussed in Section 2.2.2. When adding explainability to a development process, it must be integrated naturally into all development stages followed by a software team. Amershi *et al.* [2] describe the importance of ensuring data validity and understanding how the ML system operates. Following this guideline, I have divided the questions into four groups.

- **Model** concerns the development and understanding of the ML model and how it operates. The deep learning model is a black box and it is important to understand how it operates to identify errors and improvements.

- **Prediction** concerns understanding individual predictions of a model. Knowing why the model made a specific decision is vital for establishing trust and accountability.

- **Data** concerns understanding the dataset as a whole. Datasets are a vital component of the ML process as noted by Amereshi *et al.* [2] and Polyzotis *et al.* [43] and heavily influence the quality of an ML solution.

- **Pipeline** concerns workflow operations such as data validation and processing. Many ML systems include automated data processing and cleaning given the large quantities of data. It is important to validate how these datasets are being handled to avoid data errors and establish trust in the correctness of the system.

Using these groups, I developed a set of 11 questions to evaluate how well a tool satisfies the explainability softgoals. Each of these questions will be part of one of the four previous categories. Each of these questions can be comprehensively answered using direct information from the original paper describing the given XAI tool and an example deep learning based system.

Model Questions

1. Why does this deep learning model operate differently than an expert in the field? (Trust-ability, Disrupt-ability)
2. What is occurring in the internal workings of the deep learning model? (Trust-ability, Interpret-ability)
3. Why are specific parameters and settings significant to how the deep learning model is trained? (Trust-ability, Interpret-ability)

Prediction Questions

1. What in the input data influenced a single prediction? (Justify-ability)
2. Why should a stakeholder trust a given prediction from a model?
   (a) What can we offer to the stakeholder to build confidence in the prediction of the deep learning model?
   (Trust-ability, Explain-ability, Apply-ability)
   (b) Does this prediction confirm stakeholder expectations? If not, can the model or process show why this prediction is the most likely result?
   (Trust-ability, Disrupt-ability)

3. What risks are involved in trusting this prediction over current methodologies? (Trust-ability, Disrupt-ability)

Data Questions

1. Why are specific patterns, distributions, or slices of the dataset significant? (Interpret-ability, Apply-ability)
2. Why are specific features important to the deep learning model? (Interpret-ability)
3. Why is this dataset to be relevant and unbiased for our deep learning model? (Trust-ability, Apply-ability)

Pipeline Questions

1. How is the performance and validity of the data processing verified? (Trust-ability, Apply-ability)
2. Why are specific changes or augmentations to the dataset useful? (Interpret-ability, Apply-ability)

In order to score each of these questions, I reviewed and referenced information in each of the papers where the XAI tool was proposed. From the original, I gave each tool a subjective score between 0 and 2. A 0 represents no existing work or reasonable cause to show this methodology can answer this question. A 1 represents some work or the way to answer this question is proposed without experimental evidence. A 2 represents a great deal of existing work and support that this question can be answered with experimental evidence in the original paper. While this list is by no means exhaustive, similar to the softgoals for explainability, these questions provide a methodology to investigate how these XAI tools attempt to satisfy explainability.

3.4 Tool Categorization Results

Each of the tools being reviewed provides explainability in a different form and determines explainability in a different manner. Observing how these tools answer the 11 questions outlined in Section 3.3.1 will provide a basis for comparing and categorizing these tools in how they satisfy explainability. Figure 3.2 visualizes the scores for each of the XAI tools[1]. RuleMatrix performs best on the prediction (scoring all 2's) and data questions (scoring 1's and 2's) but performs poorly on the pipeline and model questions (scoring 0 on all except model 1). LIME performs best on the prediction and data questions (scoring all 1's and 2's in both), has mixed results for the model scoring a 2, 0, and 1, and scores zeros on the pipeline questions. LRP performs mostly well on the model, prediction and data questions (scoring mostly 1's and 2's with two 0's), and scores zeros on the pipeline questions. DeepLIFT scores best on the model questions (scoring 1's and 2's) and performs with mixed results for the prediction, data, and pipeline questions. TFX performs best on the pipeline questions (scoring all 2's), and has mixed results for the other categories, performing best in prediction with three 1's.

From these flower plots, it can be seen that RuleMatrix and LIME had similar results, LRP and DeepLIFT had similar results, and TFX had results that differed from all the other tools. This is partially due to the similarities in how RuleMatrix and LIME compute and present their results: they both use approximations and treat the original model as a black box. LRP and DeepLIFT both use a methodology based on backpropagation and are able to better analyze how the model operates. TFX is different from all other methodologies as it is designed to provide data cleaning. I expected TFX to score very well on data; however, data cleaning will not always provide insight into why a model is performing in a specific manner.

I must note that almost every XAI tool, with the exception of DeepLIFT, scored 0 for

[1] There is a comprehensive table of the results in Appendix A.1 and notes and reasoning behind scores in Appendix A.2.

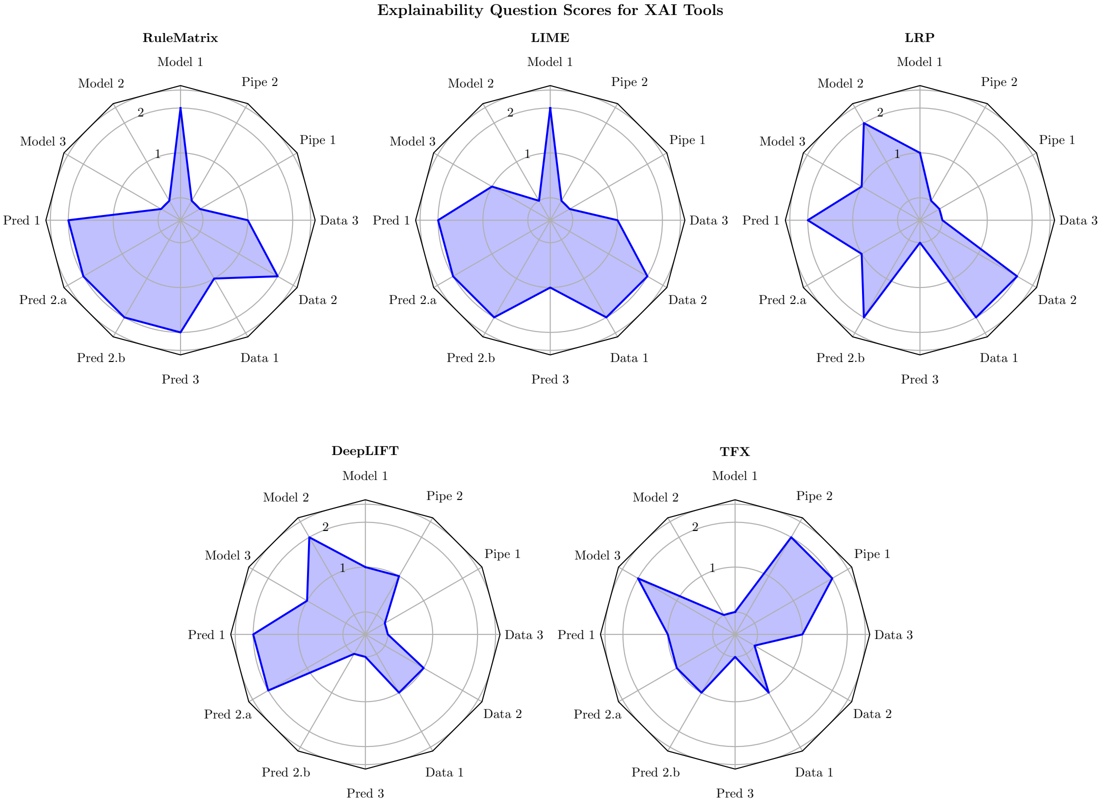


Figure 3.2: Flower plots showing how well each XAI tool answers the questions to evaluate explainability. These questions are listed in Section 3.3.1. The labels on the chart are for **Model** questions, **Pred**iction questions, **Data** questions, and **Pipe**line questions.

every pipeline-based question. These results show that most of the XAI tools do not focus on data processing or identifying sources of data errors. Data validation may not be a strong focus of these XAI tools, but since data quality plays such a prominent role, it would be useful to see how these XAI tools investigate various data augmentation methodologies. As noted by Polyzotis *et al.*, data augmentation is commonly used to improve the quantity and quality of training data [42] and it would be interesting to see if this has any effect on DeepLIFT's explainability as it was briefly explored in its original paper [50].

From these flower plots and results, almost every tool provided some insight into explaining predictions. It is interesting to note that only RuleMatrix scored a 2 when investigating prediction question 3 concerning risk. Risk is an important factor that many explanations can help mitigate when deciding between opaque ML-based solutions compared to more traditional statistical or white box solutions. Results indicate that most tools explain predictions to some degree and attempt to build trust in the predictions of the model.

RuleMatrix and LIME were also able to better identify bias (data 3) in the existing models than DeepLIFT and LRP. This may be due to the structure of the papers. When considering the entire model as a source of bias, LRP and DeepLIFT focused more on identifying how the model works, while RuleMatrix and LIME discussed the data more since they consider the model to be a black box for their analysis. From these results, there is a significant divide between RuleMatrix and LIME compared to LRP and DeepLIFT. DeepLIFT and LRP attempt to explain the model and make the model more transparent while LIME and RuleMatrix attempt to build trust in the system as a whole and the validity of the results.

3.4.1 Connecting Questions to Explainability

Results of the questions provide insight into how the tools are similar and different. However, the questions do not directly state how well each tool satisfies different identified softgoals of explainability. Despite this limitation, there are several clear trends as to how well each XAI tool satisfies each individual softgoal from this review. In the case study in Chapter 4, I will experimentally verify how well each XAI tool satisfies each softgoal of explainability.

Since these questions and their scores are subjective in nature, there is no objective way to link the questions to the softgoals. Given this limitation, it can still be seen that no individual tool satisfies all aspects of explainability but different tools provide different, partial, insights. All these tools attempt to create *trust-ability* in the model but do so in different manners. LRP and DeepLIFT provided more transparency into how the model operates such as through questions model 2. LIME and RuleMatrix provided analysis of how persuasive an answer would be to stakeholders.

Additionally, LRP and DeepLIFT provided more insight into the *interpret-ability* of the model and how it operates. RuleMatrix and LIME could provide this insight directly since they treated the model as a black box.

All of the tools scored well on the question related to *justify-ability*, prediction 1. All of these tools were designed to justify predictions of a model through either a local or global approximation and this is seen in the resulting scores. As expected, TFX scored lower than the other XAI tools as TFX is a data pipeline tool that focuses on identifying errors and not directly towards explaining a deep learning model.

For questions related to the *apply-ability* softgoal, just as with *justify-ability*, all tools built confidence that a stakeholder will trust an individual prediction (prediction 2.a). All tools can also be applied to analyze trends in the dataset (data 1). However, only LIME, TFX, and RuleMatrix were able to identify bias in the dataset. Bias is an important factor for ensuring model performance and can be identified using explainablility as shown through various studies such as Polyzotis *et al.* [43] for Google Play Store recommendations and Lapuschkin *et al.* [33] to identify biases in ImageNet [14] for identifying objects in pictures.

In addition, LIME and RuleMatrix provided support for the *disrupt-ability* softgoal scoring on all three questions related to this softgoal: model 1, prediction 2.b, and prediction 3. LRP and DeepLIFT were able to answer some of these questions but scored lower than LIME and RuleMatrix.

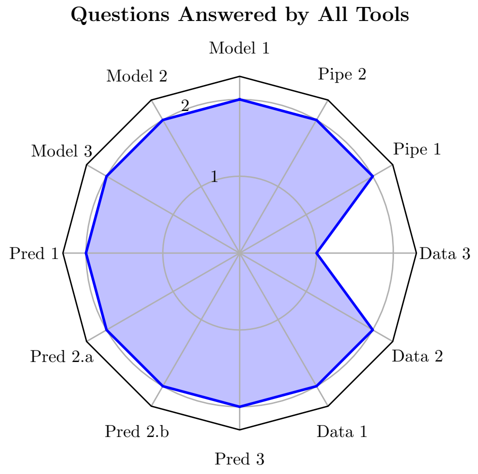


Figure 3.3: Flower plot showing the maximum score for each of the the questions across all XAI tools. These questions are listed in Section 3.3.1. The labels on the chart are for **Model** questions, **Pred**iction questions, **Data** questions, and **Pipe**line questions.

Given these results, it is clear that how an XAI tool analyzes an ML model will help satisfy different softgoals of explainability. From my review of existing tools, no one tool can solve all softgoals of explainability. Figure 3.3, a combination of the highest scores, shows that every question can be answered by at least one of the XAI tools. The only exception is risk assessment from question data 3 which can be answered but has not been investigated explicitly with evidence from these XAI tools. In Chapter 4, I further explore the application of these tools to a case study with real-world data.

3.5 Limitations of Categorization

This chapter proposes softgoals to define explainability and then applies them to existing XAI tools. The selection, scoring, and evaluation of tools is subjective. Given this subjective limitation, these questions are meant to explore what aspects of explainability each tool resolves and not to identify the best tool.

It must be noted that the tools selected were somewhat arbitrary. I selected a range of tools that provide different explanations as defined by Gilpin *et al.*'s taxonomy of explanations [17], as well as explanations using different methodologies as discussed in Section 2.3.1. This type of evaluation could be completed with other XAI tools or new tools to be proposed. I also excluded work that has been published expanding upon these tools due to limited resources to evaluate these XAI tools. This review may reveal different results when considering additional research and applications of these XAI tools. However, I expect many trends identified to remain valid since they are based on how these tools operate.

In addition to the limited selection of tools, the selected questions discussed in Section 3.3.1 are not exhaustive. More questions could be proposed to investigate different softgoals more extensively or to explore the currently identified softgoals. A unique aspect of this analysis is that it compares existing tools and establishes how they can satisfy explainability as each tool is integrated into the ML engineering workflow. Other softgoals or requirements for specific projects may lead to more questions or different priorities. In summary, the scope for this thesis is restricted to explainability to allow exploration and discussion for the case study in Chapter 4 and further discussion in Chapter 5.

Chapter 4

Case Study: XAI Tools in the Public Sector

The categorization and review of existing XAI tools provides a strong foundation for this case study. It is an experimental validation and exploration of XAI tools in a real-world application. This is an expansion on the case study submitted for publication [37].

4.1 Problem Context

Nearly 860 cities and towns in the U.S. have combined sewer systems, which manage stormwater as well as wastewater, creating what the U.S. Environmental Protection Agency (EPA) considers to be the largest unaddressed risk to human health from water infrastructure. According to a report by the EPA [57], about 850 billion gallons of untreated wastewater is discharged into waterways annually in the U.S.. This excess water from storms carries dust, trash, and debris from developed areas into the combined sewer system. When these combined sewer systems are overwhelmed, they discharge excess untreated wastewater into nearby waterways at an outfall site. This is defined as a Combined Sewer Overflow (CSO) event. Figure 4.1 shows a simplified view of the causes of CSO events.

In the U.S., there are over 9,000 outfall sites where untreated wastewater can be dis-

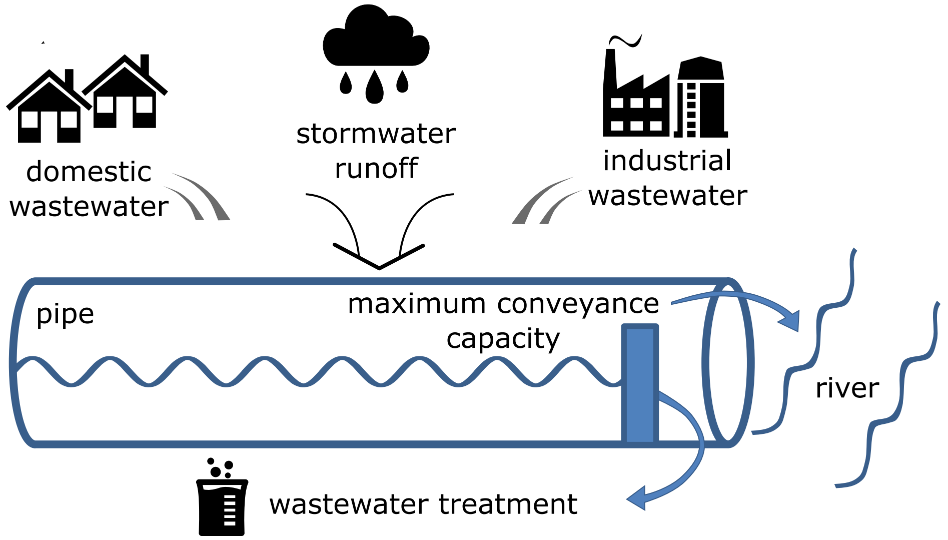


Figure 4.1: Sources of water and simplified view of CSO site and how overflow can lead into nearby water sources.

charged. Annually, CSO events cause approximately 5,000 infections, damage habitats for animals in wetlands, kill fish in rivers, and close recreational waterways and beaches [57]. This is not a problem unique to the U.S., but occurs globally. An average of 39 million tons of untreated wastewater is discharged into the Thames River in London annually due to CSO events [13]. Cities with recently constructed combined sewer systems such as Shenzhen, China, face challenges in mitigating pollution from CSO events [55]. As weather patterns become more severe due to climate change along with increasing urbanization, these problems are expected to worsen and require new solutions in the future [13].

I have been working with the Metropolitan Sewer District of Greater Cincinnati (MSD) that serves an operating area of about 290+ square miles, more than 850,000 customers, and over 3,000 miles of combined sewers [20]. MSD has recently developed a large-scale sensor network to collect data and remotely operate their system. Some of the older outflow sites can only hold a limited amount of water before they overflow and cause a CSO event. Since MSD cannot deploy any measures to mitigate CSO events at these older sites, they can only warn customers when these events occur. Their current practice is to reference weather forecasts to anticipate when CSO events occur a day in advance, then alert citizens if a CSO event is probable within the next 24 hours.

MSD is required to report and log any CSO events that occur for public safety. If they can anticipate discharges before they happen, they can mitigate risk and warn customers. They need some reasoning to justify their warning, especially when it may affect the safety of customers. Failure to warn customers and anticipate CSO events can lead to more costly repairs and risk to the local environment. This need for transparency exemplifies why relying on the weather forecast is preferred. Decisions are easy to justify, they can adjust their methodology, and this allows for future warning since rainfall and storm surges caused by rain heavily affect when CSO events will occur. Ideally, warning customers in advance can help keep them safe; however, too many false positive alerts lead to customers ignoring warnings. An alert may be sent whenever a large storm is in the forecast. Deep learning is being explored to reduce the high false positive rate in predicting CSO events ("predicting CSOs" for short).

4.1.1 Why Deep Learning

I am using a deep learning model to take advantage of a year of continuous data collected by MSD from the smart network they have developed. This dataset is a sequential, time series dataset; I am using a Long Short Term Memory (LSTM) cell structure [27] for the deep learning solution. As noted by Greff *et al.* [21], LSTM deep learning models are effective at processing sequence-based datasets and identifying relationships within the dataset. LSTMs have been applied to similar problems such as flow analysis [62]. I have explored how deep learning solutions compare to the rule-based prediction of overflow from rainfall information currently used by MSD.

4.1.2 Requirements of Explainability

Accountability is critical with any predicted CSO since MSD is accountable for their decisions. Assuming that the deep learning solution performs better than current weather

Table 4.2: Linking softgoals of explainability as defined in Section 3.1 to the CSO case study using the healthcare example discussed earlier.

Softgoal	**Healthcare Example**	**CSO Case Study**
(1) Trust-ability	Hospital staff trust the procedure to help patients.	Engineers trust the deep learning model to improve customer experience.
(2) Interpret-ability	Hospital staff interpret how the procedure will provide care to different patients.	Engineers interpret how the deep learning model processes the data.
(3) Justify-ability	The procedure and logs justify why a patient was given a specific treatment to patients and hospital staff.	The deep learning model justifies how it reached its decision to the engineers and customers.
(4) Apply-ability	The procedure applies correct decisions for patients with new symptoms or hospitals with new staff.	The deep learning model applies to new CSO sites or dataset in future for customers.
(5) Disrupt-ability	The new procedure disrupts and improves patient treatment to hospital staff experience.	The deep learning model disrupts existing methodologies for predicting CSO events.

forecasts, simply referring to a black box as having “improved performance” over other methodologies is not acceptable. When following a set of rules, the system is inherently explainable and easy to understand. However, a deep learning model is not easily understood and provides no justification to end users as to why a specific decision is reached. I applied XAI tools to a deep learning solution to satisfy the explainability softgoals.

In Section 3.2, I discussed the analogy of a hospital’s procedure for admitting patients or sending them to a waiting room and how this relates to each softgoal of explainability. An explanation of how each softgoal is related to predicting CSOs is shown in Table 4.2. The hospital analogy is included for comparison. Each of these softgoals represents an important stakeholder requirement. Satisfaction of these goals is a matter of degree with no clear-cut criterion [12].

4.1.3 XAI Tools Selection

XAI tools can be applied to satisfy these softgoals and achieve explainability of the deep learning solution. As discussed in Section 3.4.1, XAI tools satisfy these softgoals by providing explanations of predictions of the deep learning model. Figure 2.3 illustrates the process by which an XAI tool provides an explanation. Through this case study, I evaluated to what degree each XAI tool satisfies the explainability softgoals to CSOs.

As explored in Section 2.3.1, there are many different existing XAI tools that address explainability through different manners. A subset of these tools were used to determine their level of satisfaction in addressing these softgoals through the case study. Additionally, these tools are available to the engineers at MSD so they can continue to investigate these solutions without my direct input. Given this constraint, the tools should be open source, compatible with the LSTM-based solution, and easy to use.

From these requirements, I selected LIME, from Ribeiro *et al.* [47], RuleMatrix, from Ming *et al.* [40], and SHAP, from Lundberg *et al.* [36], which is based on DeepLIFT, from Shrikumar *et al.* [50]. Each of these three state-of-the-art tools can take the LSTM-based solution and provide explanations for predicting CSOs. From the scores and categorization discussed in Section 3.4.1, this case study validates the review of existing tools and how well they satisfy explainability. Each of these tools uses different assumptions and methodologies to provide explanations of predictions from a deep learning model. They are briefly described below.

- **LIME** creates a local, linear approximation of the deep learning model's output space by sampling local inputs from the dataset. LIME then uses this linear approximation of the output space to determine which features from a sample are the most significant to the prediction. Figure 4.3 shows a representation of how a linear boundary is created for a given sample of interest. This identifies the most significant features and which classes these features support. In their original work, Ribeiro *et al.* [47] applied this to an image where they could identify which pixels supported which class when identifying

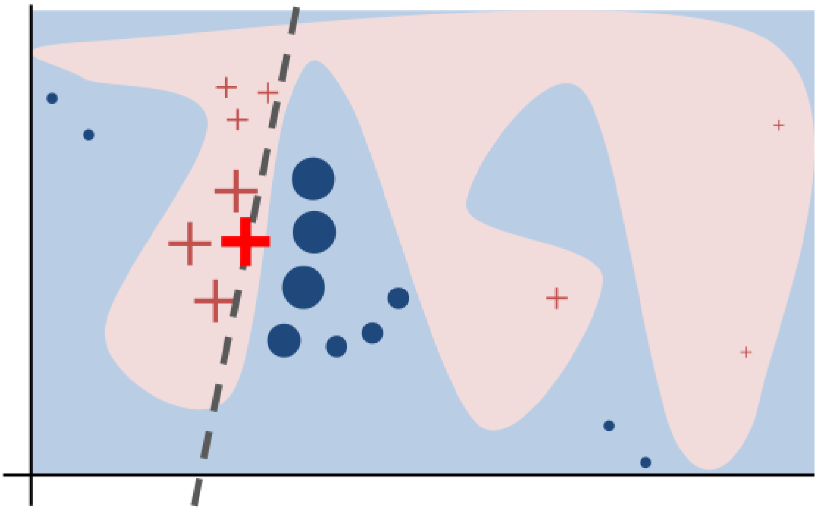


Figure 4.3: Figure from [47] showing an abstraction of how LIME forms a local, linear decision boundary from a more complex decision space.

objects within an image. LIME assigned each pixel from the input space a value of which class it supported.

- **SHAP** uses backpropagation and computes shapely values to determine how much influence the inputs of each layer have on the next layer. Starting with the prediction, SHAP progresses layer by layer from the output back towards the input, attributing influence to nodes within the network. This is used to create a significance map of how much each individual input influenced the final prediction. This methodology is based on DeepLIFT from Shrikumar *et al.* [50]; when applied to hand-written number recognition, this can produce a heatmap representing how much each pixel contributed to each class.

- **RuleMatrix** creates a global, rule-based approximation of the deep learning model from a set of given predictions. Each rule is organized hierarchically and divides the dataset using a given feature and threshold. This treats the deep learning model as a black box and uses the predictions to create a separate rule-based model. This rule-based model is considered to be inherently explainable to humans.

Table 4.4: Example of rainfall sensor data which is sampled at a rate of once per minute.

Timestamp	Rainfall (inches)
Dec 8, 2020 16:16	0.0015
Dec 8, 2020 16:17	0.0010
Dec 8, 2020 16:18	0.0006
Dec 8, 2020 16:19	0.0006
Dec 8, 2020 16:20	0.0000

4.2 Case Study Methodology

As described earlier, I developed a LSTM-based deep learning solution to predict CSOs. There are many steps to the model development as described in the ML engineering workflow from Figure 2.2. These consisted of gathering requirements from stakeholders, preparing the dataset, then developing and evaluating the model.

4.2.1 Dataset Overview

This case study focuses on data collected concerning a specific CSO site. The dataset includes various sensors at a CSO outflow site, a manhole approximately 450 ft upstream of the outflow site, and a rainfall sensor for the area. The site is considered to be "overflowing" whenever the level of water at the CSO site exceeds the site's capacity. Each site collects data independently and at different rates either every 5 minutes or every minute. In order to handle the inconsistency and variation in sampling rate and time offset, I applied linear interpolation to data collected from each source. As illustrated with fictitious values, Table 4.4 shows a sample of the rainfall data, Table 4.5 shows samples from the manhole a few minutes upstream of the outfall site, and Table 4.6 shows the synchronized and interpolated dataset from the various sensors.

Synchronizing and interpolating the dataset are vital to handle data collected at differing rates or offset in the time axis. Since I used linear interpolation, all data points are considered to form a line and missing data are sampled along that line whenever there is a missing or offset data point. This process is illustrated in Figure 4.7. As discussed with stakeholders,

Table 4.5: Example of the flow, level, and velocity data from the manhole upstream of the outfall site sampled at a rate of once every 5 minutes.

Timestamp	Level	Velocity	Flow
Aug 13, 2020 13:11:00	1.345	0.861	0.055
Aug 13, 2020 13:16:00	1.561	0.734	0.051
Aug 13, 2020 13:21:00	1.718	0.561	0.045
Aug 13, 2020 13:26:00	1.256	0.541	0.039
Aug 13, 2020 13:31:00	1.193	0.435	0.036

Table 4.6: Collected dataset of three features (flow, level, velocity) from the manhole upstream of the outfall site, one feature of the level of stored water (Outfall) from the outfall site, and one feature (Rainfall) from the rainfall sensor upstream of the outfall site. This is a sample of the synchronized and interpolated data points from the dataset. The model is given 12 hours of data at a rate of one sample every 5 minutes for the 5 features, leading to a total of 720 input parameters for the model.

Timestamp	Flow	Level	Velocity	Rainfall	Outfall
Nov 13, 2020 9:15	0.038	1.441	0.673	0.0	545.78
Nov 13, 2020 9:20	0.032	1.424	0.590	0.0	545.78
Nov 13, 2020 9:25	0.035	1.395	0.654	0.1	545.79
Nov 13, 2020 9:30	0.032	1.366	0.624	0.1	545.80

this may miss some information but should be reasonable for most types of data. An example of over-simplifying the dataset comes from rainfall which is recorded for every minute. If there is a short spike in rainfall, it may be missed by only sampling points before or after the spike in data.

While there are other methodologies of synchronizing and interpolating a dataset, such as taking the maximum over a period of time or applying a smoothing kernel, linear interpolation is one of the simplest methodologies to implement and does not cause large changes to the dataset. Additionally, this kind of interpolation allows augmentation of the dataset by shifting the starting offset and getting a slightly different set of data shifted by a small increment along the time axis. This is similar to rotating an image or flipping it across the vertical or horizontal axis.

Given the simplicity and straightforward data augmentation, I decided to use linear interpolation for this case study. Data augmentation is important to the model since only a

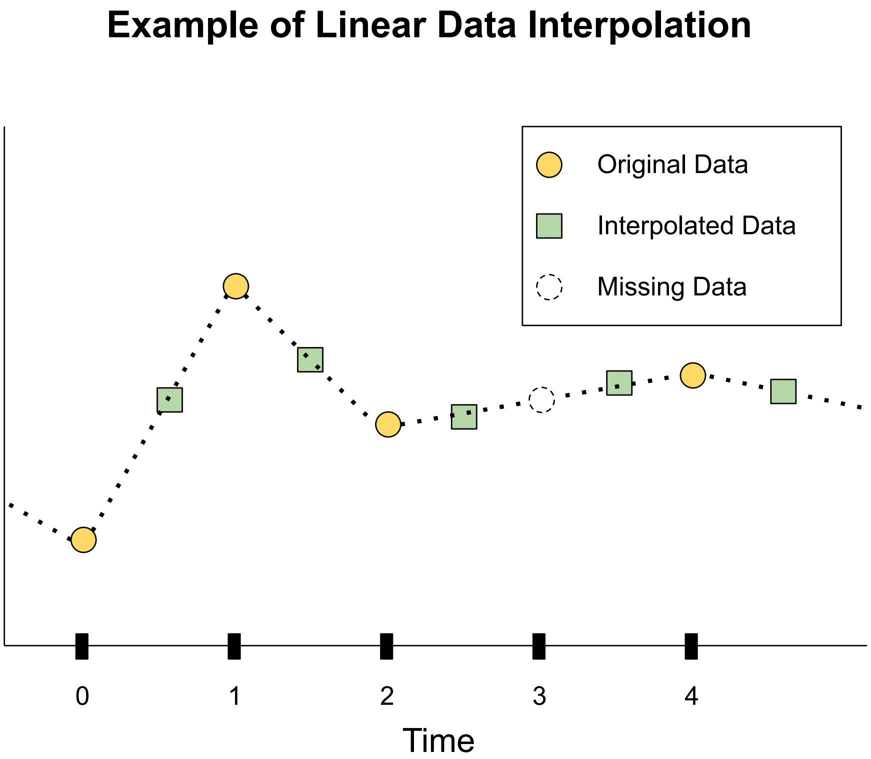


Figure 4.7: Example of how data can be interpolated using linear interpolation with examples of missing data for some values along the time axis. This can also synchronize data from various sources by sampling interpolated data from the same timestamps even if data are missing or out of sync from various sources.

limited number of CSO events occur during a given year. Using overlapping and augmented subsets of the data allowed me to evaluate and train the model on hundreds of variations of events from the dataset. These extra variations allowed for development of a more complete model despite the limitations of the dataset.

4.2.2 Deep Learning Solution

LSTMs are effective at analyzing sequence-based data as noted by Greff *et al.* [21]; given the MSD dataset, I decided to use a LSTM-based model. For this case study, I focused my resources on fully exploring the LSTM-based model instead of implementing and using other deep learning structures. In order to predict CSOs, the deep learning model reads 12 hours of data with the same structure as Table 4.6. These are then used to predict whether an CSO event will occur within the next hour. Ideally, the model should be able to identify patterns and structures within the data to predict future CSO events. The model was created using an Adam optimizer [31] and tensorflow [38] [1].

While developing the model, the CSO dataset was split into three sections: train, test, and validate. The training subset of the dataset was used to train the model; the validation subset was used to tune hyper-parameters for the model such as number of layers and nodes, and the test subset was used for final performance validation of the model. The validation and test subsets were unique as to evaluate the model on unfamiliar data and to ensure hyper-parameters were not biased towards the test subset of data. An assumption was made that each of these subsets of data were similar; a model training on the training subset should see the same type of data and situation in the test subset. Tools such as TFX [5] and data preparation helped ensure the validity of the data.

When training the model, I wanted to maximize accuracy of CSO events. The deep learning model had numerous hyper-parameters to evaluate, including number of hidden layers, size of hidden layers, and amount of data passed to the model. For each set of configuration parameters, I trained five models from a random starting state until their performance stopped improving or they had been trained for 100 iterations of the dataset. I then evaluated the results and selected the set of configurations that scored highest on the metrics of interest. hyper-parameters are important for explainability as they can affect the structure of the model and the data provided to it. Allowing for too many degrees of freedom within a model leads to low model relevance as a result of overfitting [63]. A summary of the hyper-parameters and results of training is shown in Appendix B.

There was a total of about 1000 sets of hyper-parameters were tested to find the best configuration for a validation subset of the data. This process took two weeks on a consumer level GPU: an NVIDIA GeForce RTX 2060 SUPER with 2176 NVIDIA CUDA Cores, 8 GB of RAM, 33MHz clock. Once a set of suitable hyper-parameters was identified, training the

[1] The source code and results for the case study are shared at https://doi.org/10.5281/zenodo.4579869

model took only a few hours using the GPU and tensorflow libraries. This model and final dataset were used to predict CSOs for the test data subset. Additionally, the test data and model were used to make a set of explanations for each explainability tool identified. These results are reviewed in Section 4.3.

4.2.3 Research Questions

I followed the goal-question-metric (GQM) approach from Basili *et al.* [58] and Solingen *et al.* [52] to evaluate the three XAI tools for satisfaction of explainability in the context of predicting CSOs. Previous XAI studies focused on evaluating explanations with lay persons or AI researchers [39]. Here, I collected feedback from domain experts: a hydrologist and an operational manager who work for MSD. Throughout the development of the ML solution, I communicated via email with these domain experts. In addition, we had three one-hour virtual meetings with these experts. These meetings helped to better understand the data shared, elicit explainability concerns, and explicate presentation of the results from the XAI tools.

The structure of the GQM analysis is shown in Figure 4.8 where the goal of explainability of CSO predictions is divided into multiple softgoals and metrics. The structure of this analysis is explicitly built upon human behaviour studies, making the case study directly "data-driven" according to Miller *et al.* [39]. In particular, two studies are referenced from cognitive psychology and behavioural science concerning explainability. Lombrozo [34] conducted a study with college students by posing questions and asking them to decide the most likely cause of an event from a set of choices, then justify their decision. The study concluded that students disproportionately favored simpler explanations over more likely ones when there were no probabilities given. These results indicated some preference of simplicity over correctness of a cause. Thagard [56], in developing his well-known ECHO model for representing causes for selecting between multiple explanatory processes, found that people preferred explanations more consistent with their prior beliefs. As part of this case study,

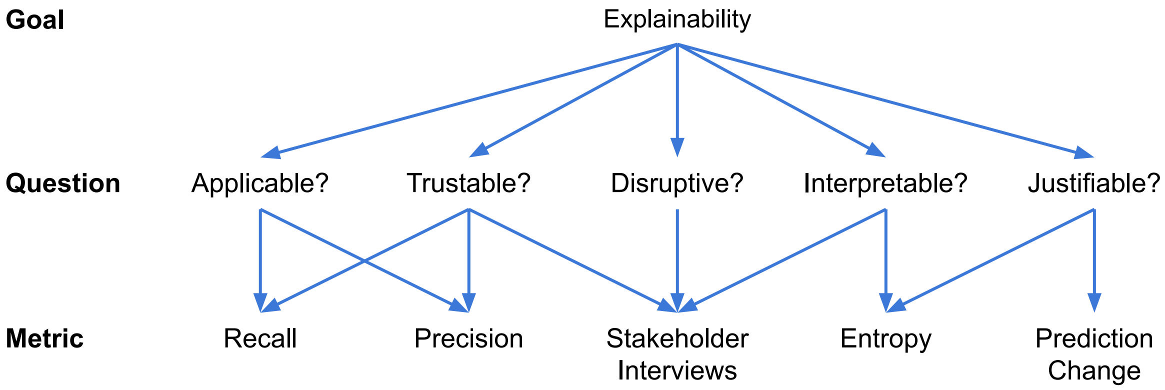


Figure 4.8: The goal-question-metric (GQM) framework guides the case study. Each arrow represents a defining concept in more detail. The diagram increases in detail from high level goals to question to measurable metrics.

I also investigated how disruptive the XAI tools' explanations are compared to domain experts' existing CSO understandings. As shown in Figure 4.8, there are five research questions related to the softgoal of explainability as described in Section 3.1.

- **RQ$_1$**: How well can the XAI-enabled deep learning solution be applied to predict CSOs?

  Stakeholders must believe the model performs well and is relevant when applied to new datasets. This explores the softgoal of *apply-ability* and confidence that the model is relevant and applicable to new situations. This was measured through *recall* and *precision* on a test subset of the dataset that the model has not seen before. Recall measures the percentage of CSO events correctly identified and precision measures the number of correctly identified events out of all predicted. Both LIME and SHAP make *post-hoc* predictions directly from the LSTM-based deep learning model. Consequently, LIME and SHAP have the same recall and precision as the LSTM. These metrics will need to be re-computed for the rule-based model generated by RuleMatrix.

- **RQ$_2$**: How much do stakeholders trust the XAI-enabled deep learning solution and its ability to predict CSOs?

  Establishing stakeholder trust is more complex than simply presenting performance metrics to stakeholders. This explores the softgoal of *trust-ability* and focuses on stakeholder opinion of the model and the explanations. Through presenting explanations and performance results from the XAI tools and LSTM-based model to our stakeholders, I assessed their level of trust in the correctness, and established transparency in how the model operates. This analysis was subjective and limited to interviews with the two domain experts from MSD. Their input was vital understanding how someone familiar with the problem assesses the correctness of the LSTM-based model.

- **RQ**$_3$: How have the explanations improved stakeholder understanding of the deep learning system?

  These XAI tools attempt to open up the black box of deep learning and provide insights into how the model predicts CSOs. This study explored the softgoal of *interpret-ability* and improved stakeholder understanding of the deep learning model. Evaluating the simplicity of explanations provides a basis for how well a stakeholder understands a given explanation. For a quantitative metric, I computed the *entropy* of the explanations produced by the XAI tools. Entropy measures the uncertainty, or disorder, of a distribution [46] which is used to approximate how much unique information is in the explanation. Additionally, I interviewed domain experts to assess how well they felt they understood the explanations and the deep learning model.

- **RQ**$_4$: How sound are the justifications provided by explanations from XAI tools?

  These explanations serve the purpose of providing accountability for using a deep learning model as opposed to current practice. Verifying the *soundness* or correctness of an explanation explores the softgoal of *justify-ability*; how well stakeholders should believe the explanations provided by the deep learning model. Soundness can be difficult to investigate since it depends on the background of a stakeholder, as discussed by Gilpin *et al.* [17]. However, XAI tools such as DeepLIFT [50], SHAP [47], and Layer-wise Relevance Propagation [4] all attempt to assess the “correctness” of the generated explanations by masking the most significant data values identified by an XAI tool from a sample to examine the corresponding prediction changes of a given deep learning model. Thus, I applied this "prediction change" metric as a quantitative measure of the soundness of an explanation.

- **RQ$_5$**: How much new insight, if any, do the XAI tool's explanations offer?

  Stakeholders at MSD want to know if deep learning can provide a new perspective on predicting CSOs. This study explored the softgoal of *disrupt-ability* and evaluated how the LSTM-based model differed from their current methodology. However, if the model deviates too much from stakeholder expectations and prior knowledge, the model may be untrustworthy. Through interviews with the stakeholders, I assessed how the deep learning model and explanations helped provide new insights to predict CSOs from the domain experts' perspective, potentially disrupting MSD's existing process.

4.3 Case Study Results

As discussed in Section 4.2.2, I developed a LSTM deep learning model to predict CSOs. This section discusses the analysis and results of the model and predictions. These results are an expansion of the results discussed of this case study that has been submitted for publication [37].

4.3.1 Model Performance

The performance of the model is significant in establishing *apply-ability* and *trust-ability* for RQ$_1$ and RQ$_2$. As discussed in Section 4.2.3, LIME and SHAP both have the same recall and precision. RuleMatrix needs to re-compute its recall and precision from its rule-based model. To calculate these metrics, both the LSTM and rule-based models were applied to the 2-month long test subset of the dataset and evaluated. The label of "elevated" means CSO events occurred; "normal" means otherwise.

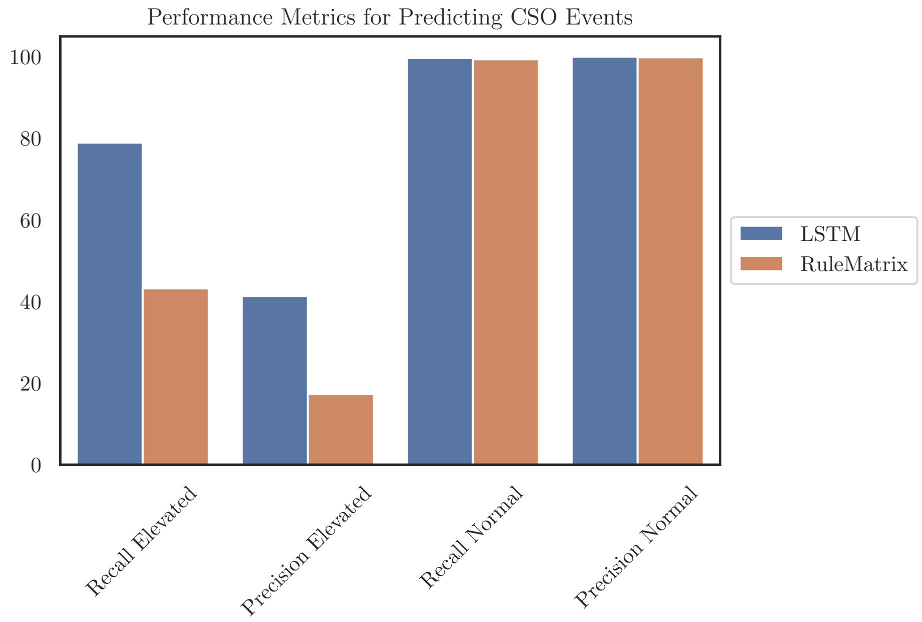


Figure 4.9: Precision and recall results for predicting CSO ("elevated" class) or non-CSO events ("normal" class). Answering RQ_1 and RQ_2 to establish performance of the model.
Recall of predicting CSOs: LIME/LSTM/SHAP=79.9%, RuleMatrix=43.2%.
Precision of predicting CSOs: LIME/LSTM/SHAP=41.3%, Rulematrix=17.3%.
Recall of predicting non-CSO events: LIME/SHAP/LSTM=99.6%, RuleMatrix=99.3%.
Precision of predicting non-CSO events: LIME/SHAP/LSTM=99.9%, RuleMatrix=99.8%.

Figure 4.9 shows performance metrics for precision and recall of both the LSTM, and rule-based model from RuleMatrix. The figure shows the results of predicting whether a CSO event will happen within the next hour for every 5 minute interval over the 2-month testing subset. The recall and precision of the deep learning model (and hence LIME and SHAP) is about 80% and 42%, respectively. The recall and precision for the rule matrix is about 44% and 18%, respectively. RuleMatrix has about a 50% drop in accuracy and 40% drop in precision compared to the LSTM on which it is based.

While disappointed with the RuleMatrix results, the two domain experts found the LSTM's CSO predictions to be encouraging and were interested in applying this methodology to more CSO sites and with more data to improve performance. During the interview, the engineers at MSD were interested in comparing the LSTM's performance to their current methodology of relying on weather forecasts to predict CSOs.

To identify their current performance, I collected rainfall data from NOAA DIVER [15] for the Cincinnati area where the CSO site is located for the same date and time range as the 2-month test dataset. This dataset collected rainfall levels recorded at a specific site daily. When using a constant threshold for a given day (0.5 inches of rainfall per day), a recall of 100% and a precision of 20% were obtained. The low precision of this methodology helped demonstrate the need for a more precise, deep learning solution. A summary and description of how the results were calculated is shown in Appendix C. Although the LSTM's 41.3% doubled the rainfall and threshold based prediction, it is important to note that the NOAA dataset only collects daily rainfall while the LSTM makes predictions continuously for every 5 minute interval. Thuts, these results do not directly compare as they operate differently and have different goals. The continuous prediction of the deep learning model may lead to lower recall, as a prediction one hour before an event may be correct but a prediction 45 minutes before an event may be incorrect; however, the event itself was still identified correctly. The LSTM accurately predicts 78.9% of the 5 minute intervals one hour before a CSO event correctly in Figure 4.9. Although the LSTM's precision suffers from the same problems as recall when continuously predicting, it still outperformed the rainfall threshold-based methodology and the LSTM should be able to be substantially improved in the future. Improvements could be developed through investigating more deep learning structures such as Convolutional Neural Networks. Additionally, the use of more datasets and longer timespans could help to make a more generalizable and robust model. The value added by the deep learning model redefines this problem and, given some improvement in performance, could be used by MSD to proactively address CSO events before they occur.

The LSTM's recall and precision of 79.9% and 41.3%, respectively, on an unseen test subset established the softgoal of *apply-ability* that the model will perform in an unbiased manner when applied to new data. This high level of performance also helped establish the softgoal of *trust-ability* to a lesser degree. Additionally, interviews with the stakeholders and reviewing their opinions of the model and the explanations helped create trust. This is discussed further in Section 4.3.3.

4.3.2 Explanation Simplicity and Soundness

RQ_3 and RQ_4 concern how the stakeholders understand the explanations provided by the deep learning model and investigate the softgoals of *interpret-ability* and *justify-ability*. As part of this analysis, I investigated to what extent "simpler explanations [are] favored over more likely ones" [34]. As illustrated in Table 4.6, the LSTM model is given a total of 720 input parameters consisting of five features sampled once every 5 minutes for 12 hours. To quantitatively evaluate the explanations provided by the three XAI tools, I generated explanations for the predictions from SHAP with all parameters, and LIME with 10 (default), 100, and 720 (all) parameters. RuleMatrix has a fixed number of decisions so all 19 rules that were produced were used in the analysis.

I used the simplicity of an explanation to measure how easily they can be understood. To quantify simplicity, I computed the entropy of explanations produced by each tool. Each prediction explanation is a matrix with the same shape as the input parameters. The influence of each element on the final prediction is stored in this explanation matrix. RuleMatrix created a set of rules. To approximate this structure, the rules were converted to a string of 1's and 0's for each rule where a 1 indicates a rule was used while a 0 indicates a rule was skipped. Since entropy measures uncertainty, a set of the same number would have an entropy of zero while a set of completely unique or random numbers would have much greater entropy. A sample of uniform random numbers was also repeated 10,000 times for each prediction size to establish a baseline of maximum entropy. Due to computational time limitations, I was only able to randomly sample a small subset of the test dataset.

Results in Figure 4.1 show an increase in entropy with an increase in explanation size (number of parameters in the explanation). This trend is expected as more unique values are being added to the explanation. For the XAI tools, LIME has a complexity almost equivalent to random sampling, and it can be filtered to find only the most significant

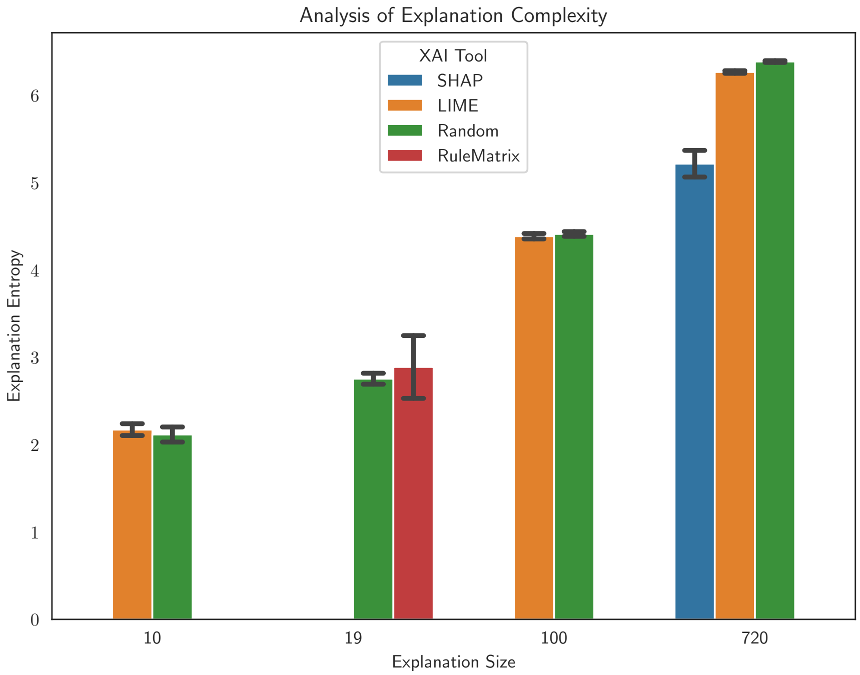


Figure 4.10: Explanation complexity measured by entropy. Random is sampled from a uniform distribution, representing a maximum baseline entropy. Entropy is computed from the Scikit-Learn library [41] with a $\log_2$ scale. Greater entropy values indicate a more complex explanation. These results evaluate how easily a stakeholder understands the explanations provided, investigating RQ_3 and RQ_4.

values. SHAP has much less entropy than the random baseline for the 720 parameters that it explained. It also has a wider variance than random sampling. This may be due to the majority of features identified as having little to no significance potentially resulting from the backpropogation methodology utilized by SHAP and the LSTM structure of the deep learning model. RuleMatrix's entropy was on par with a random sampling of a similar size but with a wider variation. RuleMatrix produced what is considered an inherently explainable model that is rule-based; that is, a human can easily understand how a given decision was reached.

Investigating RQ_3, if a stakeholder better understands an explanation, the XAI tool establishes *interpret-ability* since the stakeholder understands how the model reached a given decision. Comprehension of how a decision was reached is also important in investigating RQ_4, since a stakeholder must be able to understand an explanation to establish the softgoal

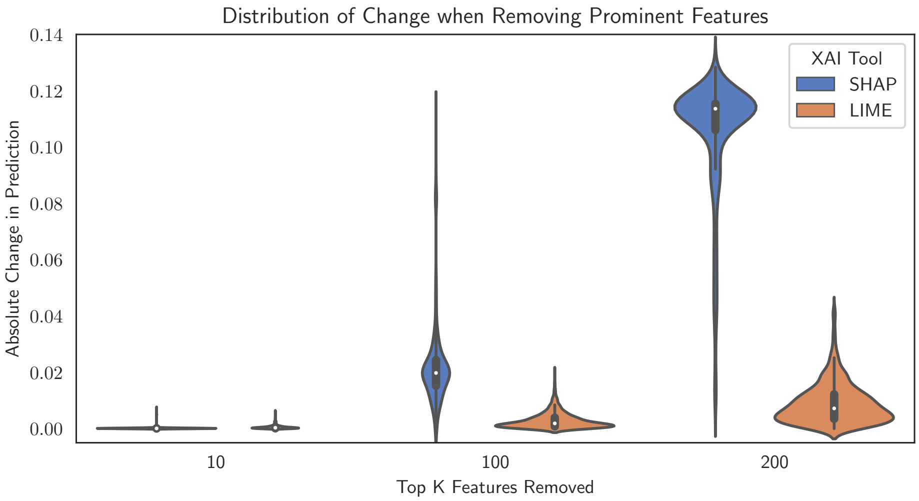


Figure 4.11: Explanation soundness generated by removing the most significant K features identified by SHAP and LIME from a sample and then evaluating how much the removal cahnged the prediction. Greater change in prediction implies a more sound explanation supporting *justify-ability* and exploring RQ$_4$.

of *justify-ability* of the model's decision. Additionally, the more confidence a stakeholder has about the correctness of an explanation, the better this explanation can be used to justify a prediction. This is quantified by computing how much the deep learning model's prediction would change when removing the most significant features at specific timestamps within the dataset. This measure provides an experimental verification of the importance of the identified features in the explanation. Since RuleMatrix did not distinguish which elements were the most significant for each individual prediction, "prediction change" cannot be directly evaluated by RuleMatrix using this procedure.

Figure 4.11 shows the "prediction change" results when removing the top K features for SHAP and LIME. For the top 10 features, SHAP and LIME have a change in prediction of about 0.0001 and 0.0007, respectively. When increasing K to 100, SHAP has a much wider range, centered at 0.0217, while LIME has an average of 0.0028. When K increases further to 200, SHAP has a significant increase in prediction change to 0.1050, while LIME only increases to 0.0088. Figure 4.11's results suggest that LIME has a slightly higher level of soundness of results for a smaller K, while a larger K shows SHAP with more sound results.

The interview with the two domain experts confirmed these observations and provided new insights. Although RuleMatrix is inherently more explainable than LIME and SHAP, the hydrologist and the operational manager at MSD did not find that the rule hierarchy captured the relevant knowledge in the CSO domain. Both LIME and SHAP were well received by the experts, with preference shifting to SHAP as this XAI tool exhibits more soundness when all features are considered together. This increased soundness created more confidence in the validity of the explanations and the justification of predictions. Despite being sound at smaller K and being flexible in terms of having a configurable number of parameters in an explanation, LIME was confusing when incorporating many features. Surprisingly, from Figure 4.10's entropy perspective, SHAP is simpler than LIME when all features are taken into account.

Framing these results in terms of RQ_3 and RQ_4, an explanation's simplicity does *not* always have to come at the cost of soundness. Although simplicity may be able to create a more easily understood result, domain experts clearly preferred soundness to simplicity. The significance of the "prediction change" metric helped to establish the softgoal of *justify-ability* through establishing belief in the correctness of an explanation and ensuring that the complexity of an explanation is understandable to a given stakeholder. Ensuring that explanations are not too complicated is important to the *interpret-ability* softgoal and allows explanations provided by a model to be understandable.

4.3.3 New Insights From Deep Learning

Interviews with stakeholders included interactive sessions where domain experts could explore XAI tools beyond the results that I had prepared. This led to a several concrete new insights focusing more on LIME and SHAP due to RuleMatrix's low recall and precision levels.

Figure 4.12 shows the general results as to which features from the dataset were the most influential to the deep learning model across all predictions. SHAP distributed influence

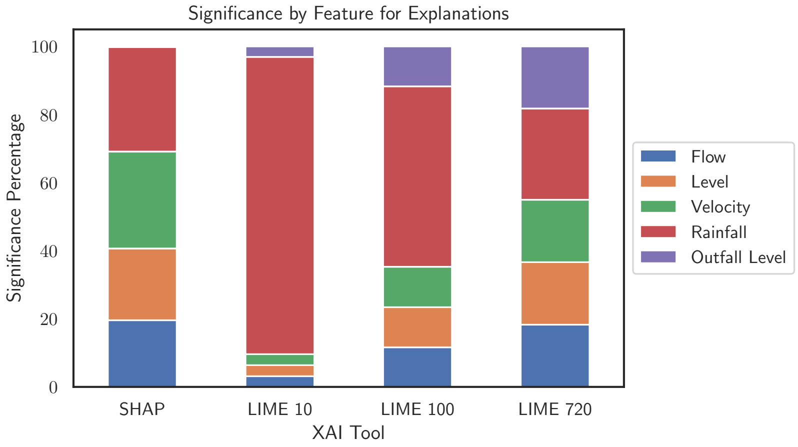


Figure 4.12: The average influence each feature had on overall prediction. Observing where the influence comes from helps establish both *justify-ability* of results (RQ_4) and explore *disrupt-ability* (RQ_5).

fairly evenly between the features while LIME heavily favored rainfall. As LIME increased to include more features in the explanation, the distribution evened out more. In addition to this, a visualization of the influence by time is shown in Figure 4.13. SHAP heavily favored the time right before the CSO event while LIME favored the start of the sample. Similar to Figure 4.12, the influence of LIME in Figure 4.13 evened out as more features were included in the explanation.

A visualization of rules produced by RuleMatrix and how samples from the dataset were divided by different rules is shown in Figure 4.14c. This visualization shows almost all the data samples are covered by a single rule of normal flow events. The other rules divided the dataset into categories by comparing a feature to a threshold at a single timestamp.

The domain experts found that LIME's results were useful in identifying the most significant features when looking at the top 10 elements. Additionally, they were able to interpret new insights from the significance plots generated by SHAP, see Figure 4.14a. They found a pattern in the correlation between velocity, flow, and level of water at the sensors upstream of the outfall site. This correlation suggested why these attributes might be more significant in a few sample events shown to them. At first, they had not assumed the feature of velocity to be useful for predictions but they realized after further discussion how correla-

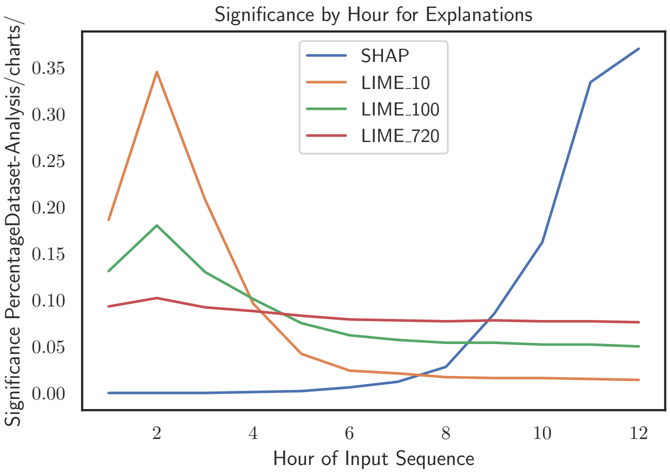
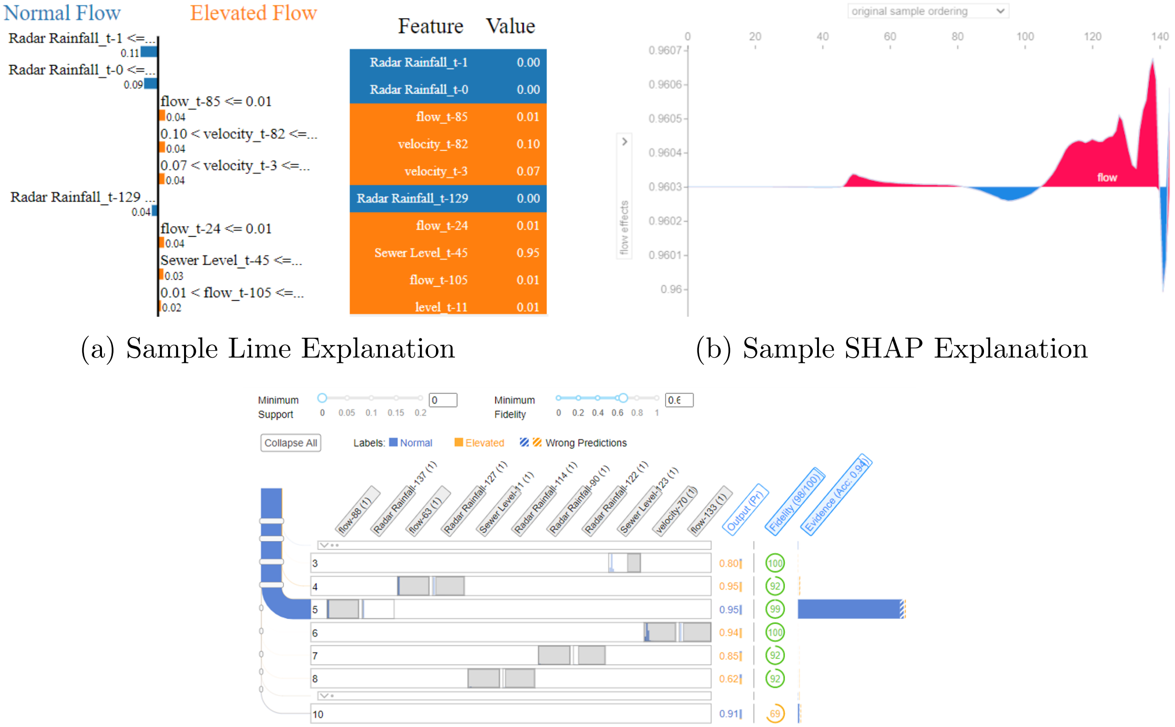


Figure 4.13: The average influence each hour of time before the predicted event had on the overall prediction. This distribution helped to identify new insights, about what specific features to focus on and *when* to focus on them (RQ$_5$).

tions between velocity and flow could help to predict CSO events in the immediate future. This aligned with the significance source being close to the time of the event for SHAP as seen in Figure 4.12. This new insight challenged their initial view and helped guide future development and resources for their analysis.

Another major insight was the significance of rainfall in the decision-making process of the model, as shown in Figure 4.12. This confirmed our stakeholders' expectations since excess storm water is the main cause of CSO events, and helped them to trust the results. However, the dominant influence of rainfall leveled off not only when LIME's explanations involved more features, but also when SHAP was applied. Given that LIME and SHAP achieved the same recall and precision levels, the patterns revealed in Figure 4.13 offer remarkable insights as to *when* to focus on which features. While the weather forecast's rainfall could still be dependent on alerting CSOs to the relevant citizens 12 hours ahead of time, paying additional, and equal, attention to sensor network data like velocity could potentially give MSD 2–4 hours, notice to dispatch engineers onsite to prevent, alleviate, or otherwise manage CSO events.

From these case study results, investigating RQ$_5$ indicated that new insights can be



Feature	Value
Radar Rainfall_t-1	0.00
Radar Rainfall_t-0	0.00
flow_t-85	0.01
velocity_t-82	0.10
velocity_t-3	0.07
Radar Rainfall_t-129	0.00
flow_t-24	0.01
Sewer Level_t-45	0.95
flow_t-105	0.01
level_t-11	0.01

(a) Sample Lime Explanation

(b) Sample SHAP Explanation

(c) RuleMatrix Visualization

Figure 4.14: Illustrating the explanations generated from the XAI tools: (a) LIME's explanation displays the most influential features supporting or opposing a prediction decision, (b) SHAP's explanation visualizes how much input features affect the CSO predictions, and (c) RuleMatrix's explanation generates a hierarchy of rules by using the input features.

drawn from the dataset concerning the softgoal of *disrupt-ability.* XAI tools, especially LIME and SHAP, have the potential to disrupt stakeholder expectations of influential factors in predicting CSOs. Additionally, these predictions can be used to take *justify-able* actions to proactively address CSO events.

4.4 Limitations of Case Study

Our inquiry was an exploratory case study [60] aimed at investigating the contemporary CSO phenomenon in depth and within its real-life context. I discussed some of the most important aspects that must be considered when interpreting our case study's results. One threat to construct validity was our choice of the three XAI tools. As mentioned in Section 4.1.3, our tool selection was guided by considerations of being open source, being compatible with our LSTM solution, and being easy for the stakeholders at MSD to use without assistance. To those ends, the scope was limited to evaluating the XAI tools *as is*, without further adjustment or customization.

Interviews with our stakeholders allowed us to identify their expectations and evaluate both the XAI tools and the analysis. It is important to note our case study's scope. I developed these metrics and conducted interviews to investigate how XAI tools can be used to evaluate a case study. I was not trying to identify the "best" tool for explainability. It must be noted that these tools are not specifically designed to visualize time-series datasets such as ours. This made the visuals and results presented to our stakeholders less informative than custom tools and visuals.

A threat to internal validity concerns the size of the real-world dataset shared with us by our stakeholder organization. Our dataset was limited in scale: the dataset was heavily biased towards the negative class (normal flow). The dataset was augmented with oversampling to effectively train the deep learning model. The augmentations might have had unintended consequences with the results of the XAI tools and this was not fully explored in our work.

More historical data from the CSO site would help evaluate our assumptions of augmentation, mitigating one threat to internal validity.

I believe our study's conclusion validity is high. First and foremost, I set out to overcome the "data-driven" weakness of current XAI studies [39]. Referring explicitly to the relevant literature [34, 56] allowed our inquires to stay focused, leading to well-grounded conclusions. Furthermore, bias was mitigated by investigating XAI tools not developed by the research team. Although two domain experts are only a small sample in the field, they have a real stake in the potential changes to be introduced by deep learning. Last but not least, I shared our source code at https://doi.org/10.5281/zenodo.4579869 in order to facilitate replication and expansion of our results.

Chapter 5

Discussion

The case study demonstrated an application of the softgoals and explored the definition of explainability. From this, I discuss the implications of explainable with respect to the case study. Then, I discuss my framework for using these XAI tools to satisfy explainability.

5.1 Satisfying Explainability

Explainability as an NFR defined in Section 2.4, is satisficed [12] in a matter of degrees. Based on the GQM analysis from Figure 4.8, I have constructed a Softgoal Interdependence Graph (SIG) in Figure 5.1. In this SIG, each of the XAI tools contributes either positively or negatively to the softgoals. Figure 5.1 illustrates that no individual tool makes all positive contributions, indicating the tools are all limited in some aspects. A tool cannot help meet some softgoal without hurting some others, suggesting some trade-off when satisfying explainability.

Explainability is divided into five softgoals defined in Section 3.1 in this SIG. As noted earlier, the performance of the LSTM and ability to predict show a significantly more trustworthy and applicable model than the rule-based model from RuleMatrix as illustrated in the SIG. The softgoals of *trust-ability* and *interpret-ability* are in contrast where when one is satisfied well the other is usually lacking. This is not inherently true of any explanation,

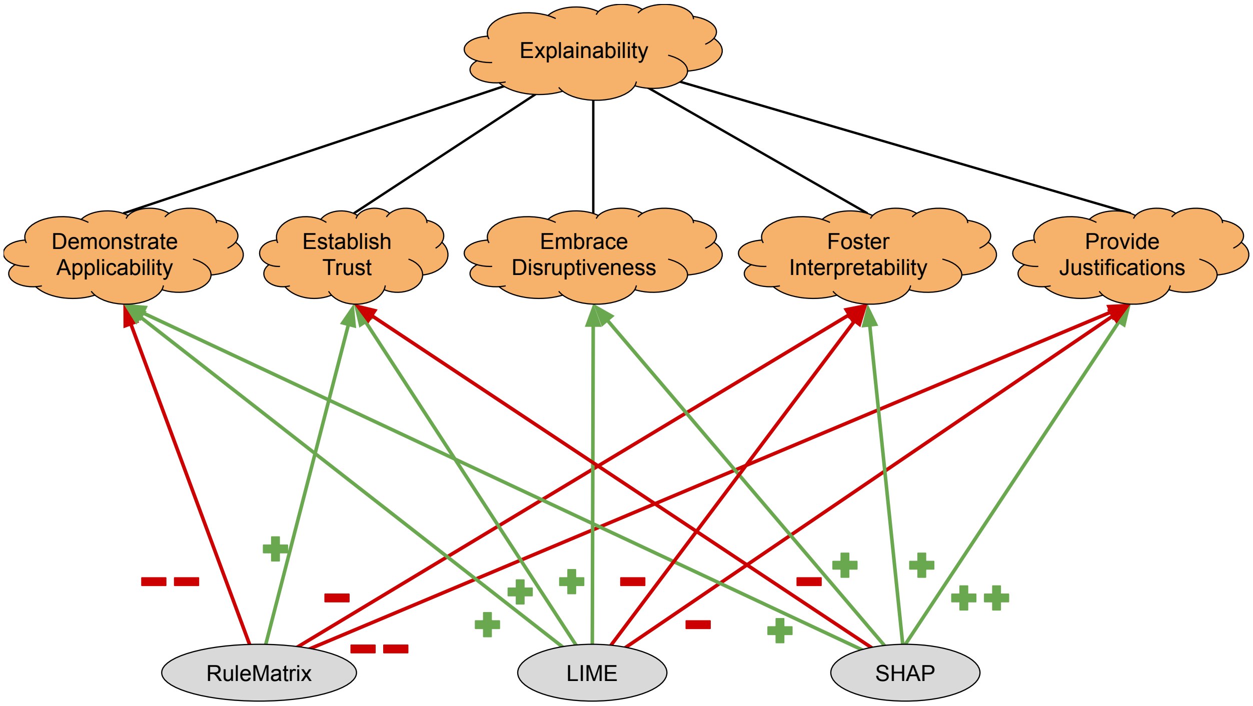


Figure 5.1: Softgoal Interdependence Graph (SIG) for XAI tools. The undirected lines represent goal decompositions, informed by the GQM analysis (see Figure 4.8). The arrows represent softgoal contributions [12]: "--" = breaks, "-" = hurts, "+" = helps, and "++" = makes.

but providing a simple, easily understood explanation makes it difficult to also give an easily interpretable explanation of the working of the model.

5.1.1 Data Driven Explainability

I specifically sought to analyze whether a trustworthy explanation has to be simple. Lombrozo [34] showed that people prefer simpler explanations over more complex ones; however, Lombrozo's study was conducted with lay persons towards a topic. As part of our case study, interviews with domain experts revealed that they greatly favored more nuanced explanations. It must be noted that simplicity does not necessarily correlate with the number of parameters explained; computing entropy to measure the randomness of information contained within explanations shown in Figure 4.10 illustrates that the *interpret-able* and *justify-able* nature co-exist within SHAP's explanations.

Thagrad [56] reported people are more likely to believe explanations that are consistent with their prior knowledge. The domain experts from MSD also showed this tendency by confirming that the XAI tools' output generally aligned with their expectations, which was positive. However, in some cases, the explanations deviated from the domain experts' prior expectations. Specifically, the results of SHAP gave more influence to velocity than the experts initially expected (see Figure 4.12); they reasoned that this explanation was taken from information they may have overlooked. Because of these observations, LIME or SHAP *alone* may not have been able to uncover these new insights. In the SIG of Figure 5.1, therefore, it is the *synergy* of LIME and SHAP that together contribute to the "embrace disruptiveness" softgoal.

5.2 Contextualization XAI Categorization

Earlier in Section 3.4.1, each XAI tool was evaluated by how well it satisfied the five explainability softgoals through evaluation of questions via subjective scores. This analysis found that there were few trade-offs when satisfying these softgoals due to the nature of how these tools operated. Different tools provided support for different softgoals. The XAI tools combined together satisfied and provided support for each defined softgoal of explainability.

5.2.1 Comparison of Case Study and Categorization

There are many parallels between the case study in Chapter 4 and the categorization of XAI tools in Chapter 3. SHAP satisfied interpretability to a greater degree than LIME and RuleMatrix in that it could provide a deeper understanding of how the model operates. However, this also caused a trade-off when establishing trust for stakeholders; more complex results are inherently more difficult to understand and do not necessarily build confidence that the model is correct. LIME and RuleMatrix satisfied trust to a greater degree than SHAP as they provided simple, more easily understood explanations as to how the model operated without further explanations. Additionally, from the case study, LIME and SHAP together helped provide *disrupt-ability*. This insight was identified when I applied these tools to real-world data and consulted stakeholders. This finding is supported by the categorization whereby both tools provided a basis for new insights.

In should be noted that, some of the findings from the case study do not directly align with the categorization. *Justify-ability* scored differently between XAI tools due to the a contrast between *trust-ability* and *justify-ability* from the SIG diagram discussed in Section 4.3.3. When evaluating these results with real data, it can be seen that the explanations provided by SHAP had far more significance on the deep learning model than those provided by LIME. Although simpler and easier to understand, explanations from RuleMatrix and LIME did not feel as sound to stakeholders when discussed during interviews. Additionally, during the case study, the bias identification did not play as much of a role in establishing explainability for LIME and RuleMatrix compared to the categorization. For the stakeholders, verifying their expectations and prior knowledge was more significant in establishing *trust-ability* than in identifying biases within the dataset. Some of these results may be due to the complex nature of the dataset, since most tools are not designed for many feature time series datasets.

Limitations of the review and categorization may have also caused some of these differences from our findings in the case study. Many of the original papers cited were published with AI researchers in mind as described by Miller *et al.* [39]. Our case study was specifically conducted with domain experts as the main stakeholders. In addition, these XAI tools and methodologies were originally designed and evaluated using image datasets or sentiment analysis and did not use sequence data such as our CSO dataset. Datasets that have complex or novel situations, such as predicting CSOs in our case study, may not be easily understood by simple models and relationships. Deep learning can help to provide new insights where other methodologies have failed. However, public services may only have smaller collections of data compared to the amount of data collected by large companies. Additionally, such as with our dataset, the data may have other requirements or intricacies that make preexisting solutions for text–or image based–datasets not applicable. However, public services have the greatest need for accountability and new perspectives on problems; our case study provides a framework to apply deep learning and and XAI tools to satisfy these requirements.

5.3 ML Engineering Workflow

An important lesson learned from our case study is that one should *not* treat explainability as something to add after a deep learning model is built. This experience advocates strongly for explainability to be engineered throughout the deep learning project. Amershi *et al.* [2] breaks the software engineering process for machine learning into nine stages. A simplified view of this process into four phases is shown in Figure 2.2. Explainability is so broadly scoped that it influenced my decisions for each phase of the case study's software engineering process.

- **Requirements Gathering**. I interviewed MSD engineers to identify what needed to be explained and why. The critical requirement of warning citizens about the CSO risks helped me to better understand the role that XAI might play in accountability, and to better build the deep learning model to make correct CSO predictions.

- **Data Preparation**. I made several assumptions about the data and applied data cleaning and augmentation by interpolating data points to synchronize the various data sources (cf. Tables 4.4–4.6). Understanding the composition of data from each of the sensors and their preparation helped better contextualize the results of the XAI tools' explanations.

- **Model Development**. Integrating XAI into the deep learning model not only required extra efforts, but also led to performance decrements as a result of the resources required to generate and visualize explanations. For tools like RuleMatrix, an additional step was required to predict CSOs according to the generated rules.

- **Model Evaluation**. Explanations can be consumed by more than just AI researchers or lay persons; from my work, I have found that domain experts can contextualize these explanations or use them to gain new insights into the task at hand. The "revision and feedback" of Figure 2.2 may involve exploring different numbers of top features from explanations provided by LIME. Additional feedback could be linked to other phases. For example, the insight gained from SHAP's results discussed in Section 4.3.3 helped elicit a new requirement of using deep learning to inform engineer-dispatch decisions 2–4 hours prior to a likely CSO event.

In conclusion, XAI tools can be integrated into all phases of Figure 2.2 as shown during our case study. They helped inform decisions throughout software development and can be integrated into pipelines as a method of verifying model performance or to diagnose issues and their causes. As noted by Zhang *et al.* [63], there is a need to identify how and why deep learning models make decisions to satisfy other broadly-scoped requirements, such as fairness, privacy, and robustness. I believe these concerns must be incorporated into all phases of machine learning development and this case study has demonstrated the feasibility of engineering explainability with state-of-the-art XAI tools.

Chapter 6

Conclusion

I presented a framework and set of softgoals for investigating explainability as an NFR. Using this framework, I reviewed different XAI tools and completed a case study to evaluate how XAI tools satisfy explainability. The definition for explainability is not conclusive and may vary for new projects with new requirements. Nonetheless, this definition provides a solid foundation for applying XAI tools to satisfy explainability.

As discussed in Section 2.2, there is a clear need for explainability in ML-based solutions and there is no clear methodology for satisfying this requirement. Large software companies including Google, Microsoft, and Amazon want explainability but do not have a clear methodology to achieve this, which is critical given the the importance of addressing policies such as the "Right to Explanation" from the European Union in the coming years. From my discussion in Section 5.1, I believe that XAI tools can satisfy the softgoals of explainability. My proposed framework and categorization provides a basis to select and evaluate tools based on a set of softgoals from stakeholders. I showed this through a case study and exploration of the NFR of accountability, which can be expanded to more softgoals in the future.

As part of our case study, I performed a GQM analysis to examine the effectiveness of explanations generated by XAI tools. I employed qualitative and quantitative methods to evaluate the XAI tools' satisfaction of the softgoals of explainability. By comparing the numeric metrics and reviewing the results of the stakeholder interviews, I was able to build upon existing psychological results and contextualize them with respect to the modern XAI tools. Domain experts welcome new insights and more complex explanations with multiple causes. Our findings do not directly refute the work of Lombrozo [34] and Thagard [56], but rather build upon their work in noting that different levels of complexity may be appropriate for different stakeholders depending on their background [17].

6.1 Recommendations for Future Work

To expand upon this work, explanations from more XAI tools can be investigated for new insights. Future work can also explore how to effectively and efficiently present value-added explanations of deep learning models to stakeholders. Additionally, more data from different sites or over a longer time period can be analyzed to expand our case study and investigate how seasonal rainfall differences affect the XAI results.

6.1.1 Explainability as Data Validation

Another further expansion would be integrating these XAI tools into the data validation proposed by tools such as TFX [5]. As part of their work, Baylor*et al.* [5] proposed a data schema to identify data errors or drift in data over time while a system is being proposed. This idea could be extended to an explanation schema to identify when the model itself changes its interpretation of the data. This would not only establish *trust-ability* but also work towards facilitating *justify-ability* by identifying where the sources of decision making of a ML-based system are coming from. As the field adapts and changes, future innovations and improved XAI tools will be able to integrate more easily into the ML workflow.

6.2 Concluding Remarks

This is a developing area; thus, the proposed GQM and data driven approach can be further improved. There will be new challenges and problems proposed by future requirements and applications of ML-based systems. Our case study shows how this framework and categorization can be applied to new challenges as new problems and tools emerge.

Bibliography

[1] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

[2] Saleema Amershi et al. "Software engineering for machine learning: A case study". In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE. 2019, pp. 291–300.

[3] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115.

[4] Sebastian Bach et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7 (2015).

[5] Denis Baylor et al. "TFX: A tensorflow-based production-scale machine learning platform". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 1387–1395.

[6] José Manuel Benítez, Juan Luis Castro, and Ignacio Requena. "Are Artificial Neural Networks Black Boxes?" In: *IEEE Transactions on Neural Networks* 8.5 (Sept. 1997), pp. 1156–1164. DOI: 10.1109/72.623216.

[7] Eric Breck et al. "The ml test score: A rubric for ml production readiness and technical debt reduction". In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 1123–1132.

[8] Emily Caveness et al. "Tensorflow data validation: Data analysis and validation in continuous ml pipelines". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020, pp. 2793–2796.

[9] Supriyo Chakraborty et al. "Interpretability of Deep Learning Models: A Survey of Results". In: *Proceedings of the IEEE International Conference on Ubiquitous Intelligence and Computing*. UIC'17. San Francisco, CA, USA, Aug. 2017, pp. 1–6. DOI: 10.1109/UIC-ATC.2017.8397411.

[10] Larissa Chazette and Kurt Schneider. "Explainability as a Non-Functional Requirement: Challenges and Recommendations". In: *Requirements Engineering* 25.4 (Dec. 2020), pp. 493–514. DOI: 10.1007/s00766-020-00333-1.

[11] Weijie Chen, Brandon D Gallas, and Waleed A Yousef. "Classifier variability: accounting for training and testing". In: *Pattern Recognition* 45.7 (2012), pp. 2661–2671.

[12] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-Functional Requirements in Software Engineering*. Springer, 1999.

[13] DEFRA. *Creating a River Thames fit for our future: An updated strategic and economic case for the Thames Tideway Tunnel.*

[14] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[15] DIVER. *Web Application: Data Integration Visualization Exploration and Reporting Application, National Oceanic and Atmospheric Administration.* 2020. URL: https://www.diver.orr.noaa.gov.

[16] Finale Doshi-Velez and Been Kim. "Towards A Rigorous Science of Interpretable Machine Learning". In: (2017). arXiv: 1702.08608.

[17] Leilani H. Gilpin et al. "Explaining Explanations: An Overview of Interpretability of Machine Learning". In: *Proceedings of the 5th IEEE International Conference on Data Science and Advanced Analytics*. DSAA'18. Turin, Italy, Oct. 2018, pp. 80–89. DOI: 10.1109/DSAA.2018.00018.

[18] Xavier Glorot and Yoshua Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. AISTATS'10. Sardinia, Italy, May 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[19] Bryce Goodman and Seth R. Flaxman. "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". In: *AI Magazine* 38.3 (2017), pp. 50–57. DOI: 10.1609/aimag.v38i3.2741.

[20] Metropolitan Sewer District of Greater Cincinnati. *About MSD*. 2020. URL: http://www.msdgc.org/about_msd/index.html.

[21] Klaus Greff et al. "LSTM: A search space odyssey". In: *IEEE transactions on neural networks and learning systems* 28.10 (2016), pp. 2222–2232.

[22] Daniel Gross and Eric Yu. "From non-functional requirements to design through patterns". In: *Requirements Engineering* 6.1 (2001), pp. 18–36.

[23] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models". In: *ACM Computing Surveys* 51.5 (Aug. 2018), 93:1–93:42. DOI: 10.1145/3236009.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. CVPR'16. Las Vegas, NV, USA, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[25] High-Level Expert Group on Artificial Intelligence, European Commission. *Policy and Investment Recommendations for Trustworthy AI*. 2019. URL: https://ec.europa.

eu/digital-single-market/en/news/policy-and-investment-recommendations-trustworthy-artificial-intelligence.

[26] Denis J. Hilton. “Conversational Processes and Causal Explanation”. In: *Psychological Bulletin* 107.1 (1990), pp. 65–81. DOI: 10.1037/0033-2909.107.1.65.

[27] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[28] Zilong Hu et al. “Deep learning for image-based cancer detection and diagnosis- a survey”. In: *Pattern Recognition* 83 (2018), pp. 134–149.

[29] International Data Corporation IDC. *Worldwide Artificial Intelligence Spending Guide*. 2020. URL: https://www.idc.com/getdoc.jsp?containerId=IDC_P33198.

[30] Anna Jobin, Marcello Ienca, and Effy Vayena. “The Global Landscape of AI Ethics Guidelines”. In: *Nature Machine Intelligence* 1 (Sept. 2019), pp. 389–399. DOI: 10.1038/s42256-019-0088-2.

[31] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[32] Maximilian A. Köhl et al. “Explainability as a Non-Functional Requirement”. In: *Proceedings of the 27th IEEE International Requirements Engineering Conference*. RE’19. Jeju Island, South Korea, Sept. 2019, pp. 363–368. DOI: 10.1109/RE.2019.00046.

[33] Sebastian Lapuschkin et al. “Unmasking clever hans predictors and assessing what machines really learn”. In: *Nature communications* 10.1 (2019), pp. 1–8.

[34] Tania Lombrozo. “Simplicity and probability in causal explanation”. In: *Cognitive Psychology* 55.3 (2007), pp. 232–257.

[35] ST John-David Lovelock, Jim Hare, Alys Woodward, and Alan Priestley. “Forecast: The Business Value of Artificial Intelligence, Worldwide, 2017-2025”. In: *Gartner.(ID G00348137)* (2018).

[36] Scott Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *arXiv preprint arXiv:1705.07874* (2017).

[37] Nicholas Maltbie, Nan Niu, Matthew Van Doren, and Reese Johnson. "XAI Tools in the Public Sector: A Case Study on Predicting Combined Sewer Overflows". submitted.

[38] Martıén Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[39] Tim Miller, Piers Howe, and Liz Sonenberg. "Explainable AI: Beware of Inmates Running the Asylum". In: (2017). arXiv: `1712.00547v2`.

[40] Yao Ming, Huamin Qu, and Enrico Bertini. "Rulematrix: Visualizing and understanding classifiers with rules". In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 342–352.

[41] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[42] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. "Data management challenges in production machine learning". In: *Proceedings of the 2017 ACM International Conference on Management of Data*. 2017, pp. 1723–1726.

[43] Neoklis Polyzotis et al. "Data validation for machine learning". In: *Proceedings of Machine Learning and Systems* 1 (2019), pp. 334–347.

[44] Mohammed Al-Qizwini, Iman Barjasteh, Hothaifa Al-Qassab, and Hayder Radha. "Deep learning algorithm for autonomous driving using googlenet". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 89–96.

[45] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016. *General Data Protection Regulation*. 2016. URL: `https://eur-lex.europa.eu/eli/reg/2016/679/oj`.

[46] Alfréd Rényi et al. “On measures of entropy and information”. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California. 1961.

[47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ““Why Should I Trust You?” Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD’16. San Francisco, CA, USA, Aug. 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[48] Sebastian Schelter et al. “Automating large-scale data quality verification”. In: *Proceedings of the VLDB Endowment* 11.12 (2018), pp. 1781–1794.

[49] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[50] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning important features through propagating activation differences”. In: *arXiv preprint arXiv:1704.02685* (2017).

[51] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: (2014). arXiv: 1312.6034v2.

[52] DM Rini van Solingen and Egon W Berghout. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. McGraw-Hill, 1999.

[53] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[54] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3319–3328.

[55] Gianni Talamini et al. "Combined sewer overflow in Shenzhen, China: the case study of Dasha River". In: Dec. 2016, pp. 785–796. DOI: `10.2495/SDP160661`.

[56] Paul Thagard. "Explanatory coherence". In: *Behavioral and Brain Sciences* 12.3 (1989), pp. 435–502.

[57] USEPA. *Report to Congress: Impacts and control of CSOs and SSOs*. 2004.

[58] Rini Van Solingen, Vic Basili, Gianluigi Caldiera, and H Dieter Rombach. "Goal question metric (gqm) approach". In: *Encyclopedia of software engineering* (2002).

[59] Meredith Whittaker et al. *AI Now Report*. 2018. URL: `https://ainowinstitute.org/AI_Now_2018_Report.pdf`.

[60] Robert K. Yin. *Case Study Research: Design and Methods*. Sage Publications, 2008.

[61] Matthew D Zeiler and Rob Fergus. "Stochastic pooling for regularization of deep convolutional neural networks". In: *arXiv preprint arXiv:1301.3557* (2013).

[62] Duo Zhang, Nicolas Martinez, Geir Lindholm, and Harsha Ratnaweera. "Manage sewer in-line storage control using hydraulic model and recurrent neural network". In: *Water resources management* 32.6 (2018), pp. 2079–2098.

[63] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. "Machine learning testing: Survey, landscapes and horizons". In: *IEEE Transactions on Software Engineering* (2020).

[64] WJ Zhang et al. "On definition of deep learning". In: *2018 World automation congress (WAC)*. IEEE. 2018, pp. 1–5.

[65] Jan Ruben Zilke, Eneldo Loza Mencıéa, and Frederik Janssen. "Deepred–rule extraction from deep neural networks". In: *International Conference on Discovery Science*. Springer. 2016, pp. 457–473.

Appendix A

Results of XAI Tool Ranking

A.1 XAI Tool Ranking Values

This appendix includes the extended justification and reasoning for each of the XAI tools scored and discussed in Chapter 3. Specifically, the results and justifications for the rankings provided in Section 3.4. The results for how well each tool answers explainability questions is shown in Table A.1.

Table A.1: How well each XAI tool answered the questions described in Section 3.3.1. A 0 indicates a tool does not clearly answer this question, a 1 indicates the tool could answer the question with no experimental evidence in the original paper, a 2 indicates that the tool can answer the question with experimental evidence in the paper.

Question	RuleMatrix	LIME	LRP	DeepLIFT	TFX
Model 1	2	2	1	1	0
Model 2	0	0	2	2	0
Model 3	0	1	1	1	2
Prediction 1	2	2	1	2	1
Prediction 2.a	2	2	1	2	1
Prediction 2.b	2	2	2	0	1
Prediction 3	2	1	0	0	1
Data 1	1	1	0	0	1
Data 2	2	2	2	1	1
Data 3	1	2	2	1	0
Pipeline 1	0	0	0	0	2
Pipeline 2	0	0	0	1	2

A.2 XAI Tool Ranking Justifications

Justifications and explanations for the scores for each tool are shown in tables by tool. RuleMatrix is in Table A.1, LIME is in Table A.2, Layer-wise Relevance Propagation is in Table A.3, DeepLIFT is in Table A.4, TFX is in Table A.5. Each of these tables contains a brief description as to why a given score was awarded based on the description and discussion of the tool in the original paper. This is a collection of my subjective notes used to decide scores for each of the XAI tools reviewed.

Table A.1: Justification for scoring of each question for the RuleMatrix tool from Ming *et al.* [40]

Question	Score	Justification
Model 1	2	A large concern of this methodology is including the opinions of experts in the field and getting their input into understanding these deep learning models and machine learning techniques.
Model 2	0	This methodology inherently changes the working of a model to be rule based and therefore will delete the original method workings. This is to say it can understanding the model as a black box but the model itself.
Model 3	0	There is some discussion but the model is considered to be an oracle so this will only ever be an approximation of the original model. There is no strong looking into what the model is composed of or why it has been trained in this manner.
Prediction 1	2	This is training a rule based model to approximate the dataset, different form other methodologies but they then try to visualize the dataset and how it makes specific predictions.
Prediction 2.a	2	They have a strong visualization process to understand which features and rules are significant (rules are an approximation of the model). Additionally, they discuss the trade-off between too much information and interpretability of the rules. An intrinsic limitation of the rule induction algorithm results from the trade-off between the fidelity and complexity (interpretability) of the generated rule list.
Prediction 2.b	2	They do not formally study how this will compare to experts as it would be on a domain by domain basis. It seems like they want their tool to be able to be used to compare with experts but it could have drawbacks being simply an approximation of the original model.

Continued on next page

Table A.1 – continued from previous page

Question	Score	Justification
Prediction 3	2	Risk is a concern and part of how they evaluated their case study when a model will effect people with possible risk factors in a medical study.
Data 1	1	They view the original model as an oracle, They can then break apart the original dataset to view how these rules are derived and where they are influenced by the original dataset.
Data 2	2	Their transformation of the dataset and model to rules does a great job of providing a justification for why features are specifically important and in what groups (that's what a rule based model is).
Data 3	1	The model's predictions are taken to be an oracle. Given this assumption it is sometimes difficult to understand differences between methodologies and understanding the validity of the dataset. Their strongly developed visualizations seem to suggest that they could be used to build trust in a model but not inherently to understand a dataset.
Pipeline 1	0	No discussion of data processing or validity in this work.
Pipeline 2	0	There is some discussion of sampling various sections of the data and this could be extended to including interpretability and comparing data augmentations but that is out of scope for the original paper.

Table A.2: Justification for scoring of each question for the LIME tool from Ribero *et al.* [47]

Question	**Score**	**Justification**
Model 1	2	Giving this a 2 because they did a good job of giving a case study of how their methodology can be compared to experts in the field and a defined methodology for applying LIME to analyze a deep learning model.
Model 2	0	Giving this category a zero they treat the model as a black box and do not focus on looking into the internal working of the model.
Model 3	1	This seems possible as they could compare multiple versions of the same model and view how the predictions change. Maybe this has been done in recent work but they did not define this well and treat the models as a black box.
Prediction 1	2	They seem to be able to develop methodologies to justify where a class gained influence for a specific prediction using their linear approximations.
Prediction 2.a	2	They developed a strong methodology for building confidence in predictions. Not going to judge the validity of this methodology merely that they proposed and executed this methodology with real humans as part of their case study.
Prediction 2.b	2	Similar with 2.a, their methodology seems to try to provide stakeholders with a way to compare their expectations to that of the dataset composition.
Prediction 3	1	This paper doesn't seem to address risk or chance of failure or looking into the correctness of a methodology compared to statics of other methodologies. I will give them a score of 1 for talking about how an end user must trust the model in order to use it in the case of a doctor as compared to traditional methodologies.

Continued on next page

Table A.2 – continued from previous page

Question	Score	Justification
Data 1	2	They seem to show how their LIME methodology can approximate overall influences for a single prediction very well and use them to compare to a human's expectations. They seems to be able to provide some global perspective on the dataset in order to establish trust. Their model is focused on local fidelity of the model but seems to be able to provide some global perspective on the model itself.
Data 2	2	Using reasoning from question Data 1, it can be extended to see that this methodology is reasonable to identify which features are globally significant and why they should be included in the model.
Data 3	1	They don't develop a clearly defined methodology but they do provide a way for evaluating the end user's trust in a dataset and apply this to given comparisons and models throughout their paper.
Pipeline 1	0	This paper does not address data processing and performance as it pertains to the model or when it could break.
Pipeline 2	0	This paper does not seem to address data augmentation or transformation and how it effects the deep learning model.

Table A.3: Justification for scoring of each question for the LRP tool from Bach *et al.* [4]

Question	Score	Justification
Model 1	1	There seems to be some evidence of this methodology being able to show where the reasoning came from for each individual category and using that to show the strength of a prediction for a particular category over other categories and where that influence came from within the model.

Continued on next page

Table A.3 – continued from previous page

Question	Score	Justification
Model 2	2	LRP gives influence of each individual node within the network on the final prediction. This may require some data processing to better understand but they are clearly tring to take apart the black box.
Model 3	1	This methodology can help for some parameters like finding which nodes are significant for making a prediction and how different nodes effect the 'relevance' of each individual node. It won't help for much with understanding how training samples influence the model but the use of backpropagation can help understand how incorrect a prediction is and visualize why that prediction is incorrect.
Prediction 1	2	This methodology does a great job of breaking down what influenced the original prediction and identifying salience from input to output space.
Prediction 2.a	1	The information given by LRP can be very complicated and difficult for a non expert to understand why this is trustworthy and they did not establish a methodology for creating trust with a user or filtering for important data. It seems like it can be done and has been partially completed but there does not seem to be a strong basis to believe this in the original work
Prediction 2.b	2	Similar to previous reasoning, but it can highlight the most important region in an image or sentence (and which parts of that region are the most influential). Given this information, I believe it is believable to conclude that this can help a user compare their expectations to that of the model given a 'saliency heatmap' over the original input dataset.

Continued on next page

Table A.3 – continued from previous page

Question	Score	Justification
Prediction 3	0	This does not help assess risk involved in methodology such as how certain these predictions or the consequences of these predictions when considering the original paper.
Data 1	2	This seems possible and may have been introduced in more recent research. You could correlate the heatmap to find significant regions or works (similar to Arras) but it has not been formalized how to go from the heatmap and original dataset to finding specific patters/distributions in the dataset.
Data 2	2	This methodology can show which features are significant to a given ML model. It has not been established in a formalized manner form this papers how to analyze the dataset as a whole and select significant features. A simple extension of the work could look into the saliency and find the on average most significant features or when features have a high correlation of 'relevance' but this is not formalized in the original paper's work.
Data 3	0	There is nothing about accounting for bias or training based in this methodology. This is simply meant to analyze an already training methodology and take it apart into pieces. There is very little concern for what the dataset that the model is trained on or how the dataset is a whole.
Pipeline 1	0	This methodology does not discuss data validity or processing
Pipeline 2	0	This does not discuss how data augmentations can effect the model or how they are significant for these predictions.

Table A.4: Justification for scoring of each question for the DeepLIFT tool from Shrikumar *et al.* [50]

Question	Score	Justification
Model 1	1	This XAI tool provides a very strong mathematical foundation for justifying the influence that a deep learning model is expected and in the SHAP [36] paper has been compared to other methodologies through changes in Log Odds ratio. But I would have to say that it does not provide an exact methodology for comparing to that of an expert in the field
Model 2	2	This methodology provides a look into how the model itself is operating using backpropagation. It seems to provide a strong justification for how the model behaves and another take on how a non-gradient based methodology can be used to identify the source of influence in the input space for a model (same with integrated gradients methodology [54]).
Model 3	1	This seems like it could be done in the future. This research in specific did not highlight looking into how groups of nodes behave and varying the node counts in networks to observe changes. I'm sure it could be done similar to LRP [4] but this research did not look into it.
Prediction 1	2	This seems to be able to successfully highlight which features can be used to make predictions for a given model and by how much.
Prediction 2.a	2	This seems to be able to successfully highlight which features can be used to make predictions for a given model and by how much.
Prediction 2.b	0	There seems to be a bit of work dedicated to parsing what is influential in the dataset and build trust that it is predicting based on the correct methodology (or at least identifying the correct source from the input data between classes).

Continued on next page

Table A.4 – continued from previous page

Question	Score	Justification
Prediction 3	0	Very little discussion of risk in trusting predictions compared to other methodologies (especially based on prediction or composition of models) .
Data 1	1	In the log odds analysis there seems to be some influence in finding patters in images that are significant to each class but there does not seem to be a dedicated effort to look further into this.
Data 2	1	This methodology seems very focused on what is important, not why it is important to the dataset or prediction. (lack of comparison between models and methodologies). There is some discussion of analyzing noise and trends in input dataset information, enough to show it is possible in the future.
Data 3	0	There is some discussion of modeling the dataset but not enough to justify how this methodology will analyze patterns, sequences, and bias of a dataset as a whole.
Pipeline 1	0	This methodology does not discuss data validity or processing.
Pipeline 2	1	There is a bit of discussion about data augmentations and processing but not really as to how it refers to the pipeline or the dataset as a whole. It could be used to compare various versions of models based on training data but that would require further research.

Table A.5: Justification for scoring of each question for the TFX tool from Baylor *et al.* [5]

Question	Score	Justification
Model 1	0	There is little to no discussion of experts in the field for specific models and identifying interpretability of a model.

Continued on next page

Table A.5 – continued from previous page

Question	Score	Justification
Model 2	0	This is focused on the information around a model, not on the model itself (with the exception of errors but that is not the focus of this work).
Model 3	2	This focuses very much so on the performance of a model and how it maintains its performance over time as the data evolves.
Prediction 1	1	There is no discussion of interpretability in this paper for influence to a model, There is a slightly tangential section focused on the predictions of a model and isolating them so there is a bit of looking into permuting the features to find errors and variations on prediction of the ml models.
Prediction 2.a	1	There is a lot of information about trusting the model but more so on the side of data validity, not what we are specifically focused on here but still related and could be extended to interpretability through a data schema and stepping through the operations of a network.
Prediction 2.b	1	This can be used to compare what a user is expecting from a template dataset but would be from a much more global perspective than that of an individual prediction. This could be applied to an individual prediction or part of the model itself.
Prediction 3	0	This focuses very much on analyzing and understanding the risk of having a change in the dataset or model with a continuous integration pipeline. But this is not the kind of risk we are looking for here
Data 1	1	This methodology focuses on where the data currently is and identifying drift and error sin the dataset itself.
Data 2	0	This methodology does not provide a basis for determining a specific feature's influence just simply that the feature is valid.

Continued on next page

Table A.5 – continued from previous page

Question	Score	Justification
Data 3	1	This is very focused on identifying statically relationships and patterns within the data and how it pertains to the predictions of the model. Maybe not the model itself but the related effects of the model.
Pipeline 1	2	Data cleaning and processing is the entire focus of this work and development with many examples.
Pipeline 2	2	This does discuss how data augmentation and modifications can be automated and verified between different steps in the process and how to identify these errors.

Appendix B

Deep Learning Training Results

This appendix contains a summary of the hyper parameter tuning and training for the deep learning model as part of the case study in Chapter 4.

B.1 Deep Learning Model Hyper-Parameters

There are several hyper-parameters that are used to tune the model. In order to evaluate the effectiveness of each set of hyper-parameters, precision, recall, accuracy, and f1 metrics were used as scores. Each of these models were trained from a random initial starting state and this process was repeated five times to account for variation in the model. The models were created using Tensorflow 2.0 and Python. A summary of the code used to create these models and tune hyper-parameters can be found at https://doi.org/10.5281/zenodo.4579869. The structure of the deep learning model itself is illustrated in Figure B.1.

Below is a summary of each of the different hyper-parameters that can be used to influence the model.

- **num_units** - Size of the hidden state of the LSTM structure of the deep learning model. More units allows for more degrees of freedom when training the model. On initial testing, values between 16 and 24 were found to have the best results when searing a large pool of possible values between 8 and 64.

- **dropout** - What percent of connections to drop when connecting between the LSTM to LSTM and LSTM to output layers of the model. This allows for avoiding overfitting while training on large complex datasets with many degrees of freedom in the model [53]. Initial testing found values between 0 and 0.5 had the best results.

- **layers** - Number of recurrent connected layers of the LSTM model. More layers allows for identifying more complex patterns within the sequence but also has the potential for overfitting to the training dataset with too many degrees of freedom. Initial testing found that either 1 or 2 layers worked best while more layers lead to overfitting or the model having difficultly converging during training.

- **class_weight** - This controls the bias given to the 'elevated' class since it is underrepresented within the dataset. Initial testing found values of either 1 or 2 had a good corrective bias without over-correcting for the lower frequency.

- **learning_rate** - This is a parameter for the Adam optimizer [31] and controls how much the model updates per training step. Initial testing found that a value of 0.001 to 0.00001 worked about the same for final performance and a value of 0.001 was used for training the model.

- **start_offset**, **end_offset** - This controls how many hours before or after the end of the input sequence the model should start predicting for overflow events. So a value of (2, 4) would me predicting overflow events between 2 and 4 hours into the future. While testing the model performed best with a smaller range close to the time of the events. Future work could look into predicting different ranges for different solutions, such as a 8-14 hour warning to deploy engineers, and a 0-1 hour warning for customers to get out of the waterway.

- **sequence_length** - How much data should be given to the model in hours. A value of 1 would indicate being given one hour of data before predicting future overflow while 48 would be 48 hours (2 days) of data before predicting. It is important to investigate as larger values may be more accurate but take much longer to train due to memory constraints on the GPU during evaluation of hyper-parameters for the model. 12 hours was found to be a good compromise between training speed and accuracy of the final model.

The metrics used for evaluating the model are listed below:

- **Accuracy** - Accuracy of the model, total number of correct predictions divided by all samples given to the model.
- **Recall-0** - Recall for class 0 (normal flow) when no overflow is occurring. Number of correctly identified normal flow events divided by *all* normal flow events.
- **Precision-0** - Precision for class 0 (normal flow) when no overflow is occurring. Number of correct identified normal flow events divided by all *predicted* normal flow events.
- **Recall-1** - Recall for class 1 (elevated flow) when overflow is occurring. Number of correctly identified elevated flow events divided by *all* elevated flow events.
- **Precision-1** - Precision for class 1 (elevated flow) when overflow is occurring. Number of correct identified elevated flow events divided by all *predicted* elevated flow events.
- **F1** - F-measure computed as a combination of Recall-1 and Precision-1 to represent how well and precisely the model predicts elevated flow. The F-measure is computed using Equation B.1.

$$\mathrm{F1} = 2\frac{\mathrm{recall} * \mathrm{precision}}{\mathrm{recall} + \mathrm{precision}} \tag{B.1}$$

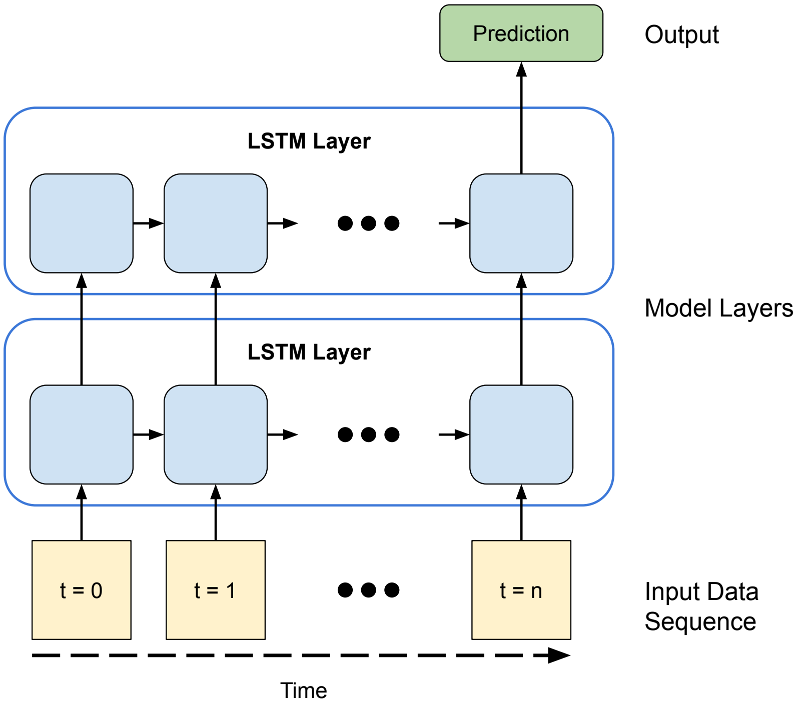


Figure B.1: Diagram representing various sections of the deep learning model and how data travels through the model.

B.2 Hyper Parameter Tuning Results

Given the various hyper-parameters for the model described earlier, the main parameters that can vary are: num_units, dropout, layers, and class_weight. Average metrics for five runs are shown in Table B.2 for each individual set of hyper-parameters. Given that some of the hyper-parameters were the same for the best models (e.g., sequence_length, learning_rage), they are not included in the table. There were more individual tests for finding acceptable ranges for each set of hyper parameter, but Table B.2 only contains the results for the final hyper parameter evaluations. From these resulting sets of hyper-parameters, the final model selected has 24 units, a dropout value of 0, using 2 recurrent LSTM layers, and a class weight of 2. The final model was training using the training and validation dataset from this set of parameters and the metrics for the final results were computed using the test subset of the dataset.

Table B.2: Resulting metric scores for each set of hyper-parameters evaluated for the deep learning model.

num_units	dropout	layers	class_weight	Accuracy	Recall-0	Precision-0	Recall-1	Precision-1	F1
16	0	1	1	0.988	0.998	0.989	0.602	0.904	0.721
16	0	1	2	0.988	0.997	0.991	0.649	0.872	0.744
16	0	2	1	0.987	0.997	0.990	0.621	0.863	0.720
16	0	2	2	0.988	0.997	0.990	0.622	0.863	0.722
16	0.25	1	1	0.988	0.999	0.989	0.571	0.933	0.708
16	0.25	1	2	0.988	0.997	0.991	0.654	0.856	0.741
16	0.5	1	1	0.987	0.998	0.989	0.573	0.886	0.692
16	0.5	1	2	0.988	0.996	0.992	0.684	0.826	0.747
16	0.5	2	1	0.988	0.998	0.990	0.627	0.892	0.735
16	0.5	2	2	0.988	0.997	0.991	0.665	0.847	0.744
20	0	1	1	0.987	0.998	0.989	0.589	0.900	0.707
20	0	1	2	0.987	0.997	0.990	0.625	0.842	0.715
20	0	2	1	0.987	0.997	0.990	0.615	0.876	0.719
20	0	2	2	0.989	0.997	0.991	0.678	0.860	0.757
20	0.25	1	1	0.987	0.998	0.989	0.577	0.915	0.706
20	0.25	1	2	0.988	0.996	0.991	0.660	0.836	0.737
20	0.25	2	1	0.988	0.998	0.989	0.602	0.918	0.726
20	0.25	2	2	0.988	0.996	0.992	0.681	0.821	0.743
20	0.5	1	1	0.986	0.996	0.990	0.610	0.834	0.698
20	0.5	1	2	0.987	0.995	0.991	0.666	0.796	0.722
20	0.5	2	1	0.987	0.997	0.990	0.620	0.845	0.712
20	0.5	2	2	0.989	0.997	0.991	0.659	0.871	0.749
24	0	1	1	0.986	0.997	0.989	0.573	0.853	0.682
24	0	1	2	0.987	0.996	0.991	0.645	0.825	0.723
24	0	2	1	0.987	0.998	0.989	0.568	0.902	0.695
24	0	2	2	0.988	0.997	0.991	0.652	0.853	0.737
24	0.25	1	2	0.988	0.997	0.991	0.658	0.844	0.739
24	0.25	2	1	0.988	0.998	0.989	0.600	0.890	0.716
24	0.25	2	2	0.987	0.997	0.991	0.643	0.847	0.728
24	0.5	1	2	0.986	0.994	0.991	0.669	0.769	0.714
24	0.5	2	1	0.988	0.998	0.990	0.610	0.900	0.726
24	0.5	2	2	0.987	0.996	0.991	0.664	0.830	0.734

Appendix C

MSD Currrent Practice

In Section 4.3.1, I compared the deep learning model to MSD's current practice for predicting CSOs. MSD does not have a log of every time they created a warning so we needed to re-compute these metrics for the given time period. In order to complete this comparison, I collected the NOAA DIVER [15] rainfall data from around Cincinnati for the daily rainfall data for the same 2-month span of the testing dataset. In order to account for missing data, I computed the maximum rainfall out of all the available sensors for each day to get an estimate of how much rain accumulated in the general area for a given day. The distribution of daily rainfall levels is shown in Figure C.1, many of the days had no (or almost no) rainfall and make up a large portion of the dataset.

To emulate sending warnings when a large storm is in the forecast, I applied different thresholds for when to warn citizens of an overflow event. If it rained more than the threshold, an overflow was predicted. There were two days with recorded CSO events for this time span. The results of computing the recall, precision, and F-measure (defined in Equation B.1) are shown in Table C.2. I selected the threshold of 0.5 inches of rainfall to compare to the deep learning methodology as it scored the highest F-measure and was able to identify all the CSO events found within the dataset.

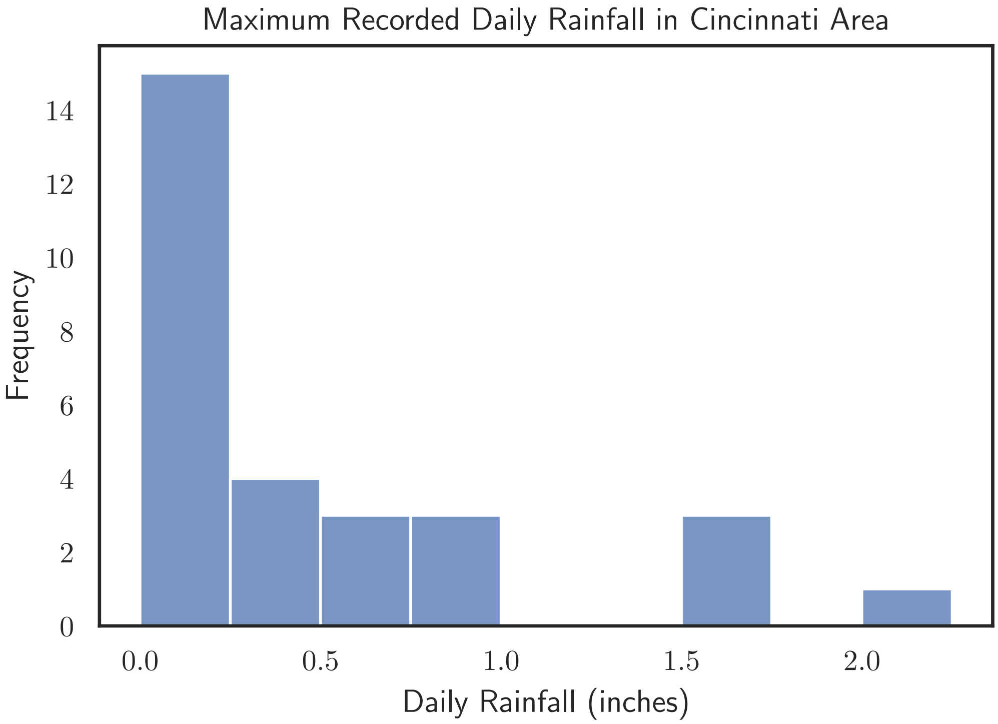


Figure C.1: Distribution of maximum daily recorded rainfall for Cincinnati area for the 2-month span of time where the testing dataset occurs.

Table C.2: Computed Recall, Precision and F1 metrics when using a flat threshold for predicting CSOs for the test dataset.

	Threshold (inches of rainfall per day)					
	0.1	0.25	0.5	0.75	1	1.5
Recall	100%	100%	100%	50%	50%	50%
Precision	13%	15%	20%	14%	25%	25%
F-measure	0.235	0.267	0.333	0.222	0.333	0.333