Process Mining/ Deep Learning Approach for Healthcare Event Prediction and Occupational Safety

by

Maryam Pishgar
M. Eng., the University of Illinois at Chicago, 2023

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering and Operations Research
in the Graduate College of the
University of Illinois at Chicago, 2022

Chicago, Illinois

Defense Committee:
Dr. Houshang Darabi, Chair, and Advisor
Dr. Lorraine Conroy, Environmental and Occupational Health
Dr. Hadis Anahideh, Mechanical and Industrial Engineering
Dr. Ugo Buy, Computer Science
Dr. Salah Fuad Issa, University of Illinois-Urbana-Champaign

ACKNOWLEDGMENTS

Before I start writing I would offer my great thanks to my adviser Dr. Houshang Darabi, for his continuous guidance, support, and inspirational mind throughout all the years. You are not only an advisor for me, you did a lot for me and I am always thankful to you. As my mother told me “You are my father in US", you changed my life in a good direction, and I owe you for my whole life. I cannot thank you enough for giving me a chance as your student. I miss talking to you about my problems and feeling much lighter when I have had a hard time. In hard times people need somebody and thanks to God that I have had you.

I would also like to extend my sincere gratitude to the rest of my committee: Drs. Lorraine Conroy, Hadis Anahideh, Ugo Buy, and Salah Fuad Issa. Your suggestions and help made the process of completing a dissertation be much more possible.

To my one and only kid, Mersana Rastegar, you are the one who has been with me from the beginning of the journey to the end! There was time that I supposed to be with you, and play! But I was busy with my research, and you! my princess, act like an adult, entertained yourself, made the environment quite so I was able to stay focused! Thank you Mersana for being the best kid in the world, for understanding and supporting me like an adult during this long journey! I love you and cannot ask God for anything more!

To my parents, Masum and Hassan. You were the only ones with me every day, for every high and low, through my whole life. Your faith in me gave me energy to do all these things. You showed me how to make a good decision in a hard situation. Asheghetoonam!

To Elham, and Mohammad my sister and brother who supported me from a far. Your warm words make me strong. You guys always tell me “We miss you."

To my friends, you all did a great thing for me. I would not have gotten through it without you all, nor would my time at Chicago and in Irvine have been as much fun. Thank you, Saif, Karuppiya, Hamza, Martha, Drs. Greg V Placencia, Mehrdad Aliasgari, Venu, Fazle Karim, and Sina Rastegar.

Contribution of Authors

The research reported in this dissertation was partly supported by the National Institute for Occupational Safety and Health (NIOSH) under the grant "Occupational Safety and Health Education and Research Centers (T42)", award number **T42OH008672**. This includes contributions from multiple papers and authors which are detailed below:

REDECA: A Novel Framework to Review Artificial Intelligence and Its Applications in Occupational Safety and Health by M. Pishgar (MP), S. F. Issa (SFI), M. Sietsema (MS), P. Pratap (PP), and H. Darabi (HD). Conceptualization, H.D. and M.P.; methodology, S.F.I., M.P. and H.D.; software, M.P. and M.S.; validation, M.P., S.F.I., M.S. and P.P.; formal analysis, M.P., S.F.I. and H.D.; investigation, all authors; resources, all authors.; data curation, M.P. and S.F.I.; writing—original draft preparation, all authors; writing—review and editing, all authors; visualization, M.P., M.S. and S.F.I.; supervision, H.D.; project administration, H.D.; funding acquisition, P.P. and H.D. All authors read and agreed to the published version of the manuscript.

Prediction of Unplanned 30-day Readmission for ICU Patients with Heart Failure by M. Pishgar (MP), J. Theis (JT), M. Del Rios (MR), A. Ardati (AA), H. Anahideh (HA) & H. Darabi (HD). MP and HD: Involved in all aspects of this study. JT: Data acquisition, data interpretation, drafting of the manuscript. MR and AA: Data interpretation and drafting of the initial manuscript. HA: drafting of the initial manuscript. All authors read and approved the final manuscript.

Process Mining- Deep Learning Approach to Predict Survival Outcome for COVID-19 Patients by M. Pishgar (MP), S. Harford (SH), J. Theis (JT), W. Galanter (WG), J. M. Rodríguez-Fernández (JMR), L. H Chaisson (LC), Y. Zhang (YZ), A. Trotter (AT), K. M. Kochendorfer (KMK), A. Boppana (AB) & H. Darabi (HD). MP, SH, HD, JT: Involved in all aspects of this study. WG, JMR, LC, YZ, AT, KMK, AB: Data acquisition and interpretation, and revision of the manuscript. MP, SH, JT have equal contribution in this paper. All authors read and approved the final manuscript.

Process Mining Model to Predict Mortality in Paralytic Ileus Patients by M. Pishgar (MP), Martha Razo (MR), Julian Theis (JT), and Houshang Darabi (HD). MP was involved in all aspects of this study; JT conceived and designed the work, and conducted data collection, preparation, and analysis; HD conceived and designed the work, conducted data analysis, interpretation, and verification, and critically revised and gave final approval of the article; MR conducted the write up.

Improving Process Discovery Algorithms Using Event Concatenation by M. Pishgar (MP), Martha Razo (MR), and Houshang Darabi (HD). MP and HD were involved in all aspects of this study; MR conducted the write up.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

SUMMARY

With the ongoing digitization of industries and the rising number of interconnected devices, an increasing amount of system recordings are created. Process mining describes a set of techniques that help in the automatic discovery of mathematical models from such system recordings. Such system recordings could be complex including many concurrencies, noisy and infrequent behaviors. Even though many process discovery algorithms and pre-processing steps have been proposed to remove such behaviors, still none of them have been able to decrease the complexity of the process models and show the dynamic behavior of the system closely. The analysis of the discovered models might reveal important knowledge about the system behavior that is otherwise impossible to obtain. Therefore, generating a pre-processing step to improve the quality of such recording, hence improving the quality of the process model is essential.

This dissertation first focuses on generating a pre-processing step, concatenation algorithm, to improve the quality of the system recordings, consequently, the process model, and improving the results of the evaluation metrics. This algorithm finds all possible combinations of concurrent events and selects several of these combination for concatenation based on a probability function, then removes all the self-loops. Significant improvements have been observed by applying concatenation algorithm on 18 benchmark datasets.

Process mining techniques mainly rely on discrete event system theories in constructing the corresponding mathematical models. Therefore, they perform poorly, when continuous measures such as time and probabilities of the system events need to be considered. On the other hand, such measures can be effectively modeled and predicted through deep learning. Hence, a system that could leverage both process mining and deep learning techniques for the prediction would be useful and effective.

A second contribution applies a process mining/deep learning model formalism to predict certain behaviors in healthcare systems of three real world case studies. The detailed models and the results are explained in this dissertation. For this contribution, Electronic Health Records (EHR) are first converted to the event logs. The event logs are then fed to the process discovery algorithm to produce the process model. Finally, the resultant process model, the event logs, demographics of the patients, and severity scores are fed to Decay Replay Mining (DREAM) algorithm to predict outcomes and the prediction model is evaluated through several common metrics. Significant improvements have been observed using DREAM algorithm for prediction.

The third contribution of this thesis is to apply the concatenation pre-processing algorithm to real-life healthcare datasets to demonstrate its effectiveness on complex healthcare datasets. The real-life healthcare datasets are complex and noisy. Infrequent behaviors and various concurrences exist in such datasets, which causes generating inefficient and complex process model through process discovery algorithms and leading to inaccurate predictions. Therefore, it is critical to pre-process raw data to improve its quality, hence making the process model more understandable and critical predictions more accurate and trustworthy to the medical team.

For this contribution, EHR are first converted to the event logs. The concatenation algorithm is then applied on the event logs. The resultant event logs are then fed to the process discovery algorithm to produce the process model. The process model is then evaluated by using common evaluation metrics. Finally, the resultant process model, the event logs, demographics of the patients, and severity scores are fed to the DREAM algorithm to predict the critical health outcomes and the prediction model is evaluated through Area Under the Curve (AUC) metric. The same procedure is tried by using raw event logs and the results are compared. Significant improvements have been observed applying concatenation as a pre-processing step and DREAM algorithms as a prediction model.

The field of Artificial Intelligence (AI) is rapidly expanding, with many applications seen routinely in health care, industry, education, and increasingly in workplaces. Although there is growing evidence of applications of AI in workplaces across all industries to simplify and/or automate tasks there is a limited understanding of the role that AI contributes to address Occupational Safety and Health (OSH) concerns. Hence, a framework which reviews the application of AI in workplaces is essential to highlight the role that AI plays anticipating and controlling exposure risks in a worker's immediate environment.

A fourth contribution is the creation of a framework to review the application of AI in OSH in five main industries. Risk Evolution, Detection, Evaluation, and Control of Accidents (REDECA) describes the potential applications of AI in anticipating and controlling occupational hazards and opportunities for future AI interventions. The REDECA framework has also been used to identify the existing AI solutions and specific areas where AI solutions are missed and can be developed to reduce incidents and recovery time for the agricultural tractor drivers. In this work, 260 AI papers across five sectors (oil and gas, mining, transportation, construction, and agriculture) are reviewed using the REDECA framework to highlight current applications and gaps in OSH and AI fields. As a result of this study, most of the evidence of AI in OSH research within the oil/gas and transportation sectors focuses on the development of sensors to detect hazardous situations. In construction the focus is on the use of sensors to detect incidents. The research in the agriculture sector focuses on sensors and actuators that removes workers from hazardous conditions. Application of the REDECA framework highlights AI/OSH strengths and opportunities in various industries and potential areas for collaboration.

The fifth and final contribution is the case of agriculture which has the highest rates of fatality incidents in the US, 600 cases in 2019, the numbers of the tractor-related injuries have remained high with 213 cases from 1999 to 2019. Therefore, identifying root causes of agricultural tractor driver incidents, determining existing AI solutions to reduce the agricultural tractor driver incidents is needed to improve the safety of tractor drivers in the future.

In this work, 171 Fatality Assessment and Control Evaluation (FACE) reports related to the tractor drivers and REDECA are used to identify the existing AI solutions and specific areas where AI solutions are missed and can be developed to reduce incidents and recovery time. Fatality reports of tractor drivers are categorized into six main categories including: run over, pinned by, fall, others (fire and crashes), roll over and overturn. Each category was then subcategorized based on similarities of incident causes in the reports. As a result, the application of the REDECA framework has revealed potential AI solutions that could improve the safety of tractor drivers.

INTRODUCTION

1.1. Process Mining and the Need to Pre-Process before the Process Discovery

Process Mining is a comparatively young research discipline that aims to discover and analyze processes by extracting knowledge from recordings that were obtained during the runtime of a system. Such recordings consist of sequences of instantaneous events that are associated with timestamps of occurrence and optional metadata. However, the sequence of the recordings is very noisy, and contain infrequent behaviors and concurrencies. Process models can be extracted through process discovery algorithms to illustrate and discover contextual relationships. Most of the process discovery algorithms use all available recordings to produce a process model which mostly leads to a complex process model with a lower quality that shows fewer real behaviors of the recordings. The main objective is to discover a process model that reflects the true underlying system behavior and is simple enough to be understandable. As a result, there is a need of a pre-processing to remove infrequent behaviors and the concurrencies consequently a simpler process model will be produced.

1.2. Process Mining and Deep Learning

Operational support is another process mining discipline which focuses mainly on predictive objectives during runtime, such as future behavior. Various systems and processes are deployed across all industries follow predefined strategies that maintain core aspects such as standards, guidelines, and proper business executions. When systems malfunction, organizations might be confronted with a loss of revenue, reputation, workforce, or assets. Therefore, there is a strong interest in operational support that enables intervention and assists in resource allocation and/or planning. Deep learning has gained acceptance predicting system behavior during runtime due to the modeling power of neural networks. Prediction of future events and time properties are two examples. However, pure deep learning methods loses the advantages gained with process mining. These advantages include the discovery of interpretable process models used to reason about a prediction and obvious logical rules of the underlying system that must be partially relearned by the deep learning algorithm. A symbiosis of process mining and deep learning techniques to predict system behavior therefore combines the advantages of both disciplines, that can have many significant real-world applications, such as in the healthcare industry. Healthcare providers are interested in accurately predicting future patient critical outcomes such as mortality and hospital readmission, unveiling missing health record information, and detecting potential fraud or reporting errors. Combined process mining and deep learning methodologies will enable them to predict accurately such outcomes of interest and obtain process models that reflect the underlying system behavior beyond recordings. This will improve in-depth analysis of the healthcare system.

1.3. Application of Pre-processing to Predict Critical Healthcare Outcomes

Infrequent behaviors and various concurrences make real-life healthcare datasets complex and noisy. This generates inefficient and complex process model found by process discovery algorithms leading to inaccurate predictions. Therefore, it is critical to pre-process raw data to improve its quality, hence making the process model more understandable and critical predictions more accurate and trustworthy to the medical team.

1.4. Application of Artificial Intelligence in Occupational Safety and Health

AI is the study of agents that take data from the environment, analyze it, and act based on the analysis. The process is initialized by collecting data from the environment by sensor devices, analyze it using machine learning algorithms, then use actuators to perform actions. Sensor devices and actuators are autonomous parts of AI, while machine learning techniques are the algorithmic part of the AI. Since the application of AI in workplaces has increased over the past few years, it is very critical to understand AI methods thoroughly, and their effects on safety improvements of the workers and workplaces. Hence, a framework that reviews the application of AI in the workplaces in main industries and finds gaps where AI could be useful is needed.

1.5. Contributions

The first contribution of this dissertation is a pre-processing step that concatenates events holding concurrent relations, causing the higher quality process model to be generated by process discovery algorithms. The resultant process model is simpler and shows closer behaviors of the event log as compared to using the raw data for process discovery. Significant improvements were seen in the evaluation metrics on 18 benchmark datasets after applying the pre-processing step.

The second contribution of this dissertation is an approach to accurately predict the important health outcomes by combining the advantages of process mining and deep learning techniques. The approach first constructs a process model by using process discovery algorithm. The model is enhanced with time decay functions to create continuous process state samples. These samples are combined with discrete token movement counters, Petri Net markings, demographic information of the patients, and severity scores related to each patient to train a deep learning model that predicts the health outcomes. These methods outperformed current state-of-the-art methods on three case studies. By following this approach, the time information related to the variables and the history of the patients are retained as inputs to the deep learning model for training. Furthermore, this approach retains an explicit process model compared to pure deep learning models that increase their interpretability.

The third contribution of this thesis applies concatenation pre-processing to real-life healthcare datasets to demonstrate its effectiveness. EHR are converted to the event logs and fed to the concatenation algorithm. The resultant event logs are then fed to the process discovery algorithm to produce the process model. The process model is then evaluated by using common evaluation metrics. Moreover, the resultant process model, the event logs, demographics of the patients, and severity scores are fed to the DREAM algorithm to predict the critical health outcomes and the prediction model is evaluated through AUC metric. The same procedure is tried by using raw event logs and the results are compared. Significant improvements have been observed applying concatenation and DREAM algorithms.

Another contribution introduced a new framework named REDECA, to review the application of AI in OSH. It also identified research studies that highlighted current applications of AI to improve the health and safety of workers in agricultural, oil and gas, mining, transportation, and construction industries. Moreover, it described potential applications of AI in anticipating and controlling occupational hazards, and opportunities for future AI interventions.

Lastly, we identified root causes of agricultural tractor driver incidents, determined existing AI solutions to reduce the agricultural tractor driver incidents to improve the safety of tractor drivers in the future. 171 Fatality FACE reports related to the tractor drivers and REDECA are used to identify the existing AI solutions and specific areas where AI solutions are missed and can be developed to reduce incidents and recovery time. Fatality reports of tractor drivers are categorized into six main categories including: run over, pinned by, fall, others (fire and crashes), roll over and overturn. Each category was then subcategorized based on similarities of incident causes in the reports. As a result, the application of the REDECA framework has revealed potential AI solutions that could improve the safety of tractor drivers.

1.6. Outline

This dissertation is structured as follows. Chapter 2 provides a background review by introducing required definitions, process models, process mining, deep learning, specific algorithms, statistical tests, and relevant software. Chapter 3 proposes a pre-processing step to improve the results of the evaluation metrics. Chapters 4 contains three subsection which are 4.1, 4.2, and 4.3. All three case studies use the deep learning/ process mining approach to predict important healthcare outcomes. The outcomes, types of diseases and patients are different in each subsection. Chapter 5, we applied concatenation algorithm to predict critical health outcomes. Chapter 6 reviews the application of AI in OSH. Furthermore, Chapter 7 used the REDECA framework to review existing AI solution for the tractor drivers and reveals potential AI solutions that could be used to improve the safety of tractor drivers. Chapter 8 proposes future work and conclusion.

BACKGROUND REVIEW

1.1. Definitions

Definition 1 An event $a \in A$ describes an instantaneous change of state of a system S. A is the finite set of all possible events. A specific event a may happen more than one time during the runtime of a system [1].

Definition 2 An event instance E is a vector with at least two attributes: the associated event a and the corresponding occurrence timestamp. An instance vector may contain further non-mandatory attributes like costs, people, and resources associated with that event occurrence [2]. Two event instances cannot have the same timestamp, i.e. cannot occur simultaneously. This is because of the continuous nature of time, and the fact that point probabilities in continuous probability distributions are zero. N defines the set of all possible event instances and D is the set of all possible attributes. Then for any event instance $E \in N$ and any attribute $d \in D$: $\upsilon_d(E)$ is the value of the attribute d for the event instance E. If an event instance E does not contain an attribute d, then $\upsilon_d(E) = \emptyset$ (empty set). The timestamp attribute is denoted by d_{ts} [1], [3].

Definition 3 A trace $c \in C$ is a finite and chronological sequence of event instances. C defines the finite set of all possible traces. The function $\gamma(c)$ returns the number of event instances in c, i.e. the length of the trace. The ith event instance of c is denoted by c_i [1], [4].

Definition 4 A variant $v \in V$ is a sequence of events where V refers to the infinite set of variants. The function ω: $C \to- V$ maps traces to variants such that for a given c its variant satisfies $\forall 1 \leq i \leq \gamma(c) : vi = \upsilon_{event}(ci)$ [1], [4].

Definition 5 An event log $L \subseteq C$ is a set of traces where $L_{i,j}$ refers to the jth event instance in the ith trace of the event log. $|L|$ denotes the cardinality of L corresponding to its number of traces. The function $\gamma(Li)$ expresses the number of event instances of the ith trace of the event log L [5].

Definition 6 A variant log L^* is a sample of variants of size $|L|$ and defined such that $\forall_{c\in L}$: $\omega(c) \in L^*$ and $\forall_{v\in L^*}\ \exists_{c\in L}$: $v = \omega(c)$ [1].

Definition 7 A unique variant log L^+ is the set of variants contained in L^*, i.e. $\forall_{v\in L^*}$: $v\ \exists!\ L^+$. Let R be a random variable of S that takes on variants $v \in V$ and follows a probability density denoted by P. When observing S and recording an event log for an infinite period of time $t \rightarrow \infty$, the relative frequency of each $v \in L^*$ will follow P [5].

Definition 8 Directly-Follows Frequency, given any event logs L and any two event labels L, $L' \in L$, the directly-follows frequency of $L \rightarrow L'$ denotes the number of times L' immediately appears after L, in at least one trace in the given event logs L. We denote this frequency with $|L \rightarrow L'|$ [1], [6].

Definition 9 A self-loop exists in our event log L, if $|L \rightarrow L|$ is positive for some event $a \in A$ with $\lambda(a) = L$ [7].

Definition 10 Concurrency Relation. Given any event logs, L and any two labels L, L', and any two events $a_i, a_j \in A$, are said to have concurrent relation, denoted by $(a_i \| a_j)$, if and only if the following conditions exist: $|L \rightarrow L'| > 0$ and $|L' \rightarrow L| > 0$ where $\lambda(a) = L$ and $\lambda(a') = L'$ [1], [7].

1.2. Process Models

Process models are used to reason about processes and systems and are therefore a core component of process mining. Such models are supposed to provide insight into the control-flow perspective, i.e. they unveil in which order events $a \in A$ are performed and detect the logical patterns which a process follows. Such process models are usually in the format of Petri Net [8].

1.2.1. **Petri Nets**

A Petri Net (PN) is a mathematical model that can represent a process. It consists of a set of places; these are graphically represented as circles and transitions represented as rectangles. Transitions correspond to events. Transitions and places are also referred to as nodes. Additionally, arcs are used to unidirectionally connect places to transitions and vice versa. A labeled PN is defined as:

$$PN = (P, T, F, A, \pi) \tag{2.1}$$

where P is the set of places, T is the set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs connecting places and transitions, and A is the set of events [4]–[6]. The set $P \cup T$ is called the set of nodes. The first node of each pair $(x, y) \in F$ represents always the source whereas the second node represents always the sink of the directed arc. In other words, a node x is the input node to another node y iff $(x, y) \in F$. Similarly, x is the output node to another node y iff $(y, x) \in F$. For any $x \in P \cup T$, $\bullet x = \{y| (y, x) \in F\}$ is the set of input nodes to x and $x\bullet = \{y| (x, y) \in F\}$ is the set of output nodes of x. The function π: $T \rightarrow A \cup \{\perp\}$ maps each transition $t \in T$ to either a single event of A or to the non-observable event $\perp$. A labeled PN is defined such that

$$\forall_{a \in A} \exists_{! t \in T}\ \pi(t) = a \tag{2.2}$$

Each place can hold a non-negative integer number of tokens. The function $\sigma(p)$ returns the number of tokens in a place p where $p \in P$.

The state of a PN corresponds to a marking $M \in M$ where M is the set of all possible markings. One defines $M \in Z^{|P|}$ as a vector of size |P| where Z denotes the set of all non-negative integers and |P| corresponds to the cardinality of P. Each element $Mi = \sigma(p_i)$, i = 1, ..., |P| where p_i

is the ith place of P. The initial state M^{init} is also called initial marking, whereas the final state M^{final} is called final marking. Usually, process models have a dedicated source and a dedicated sink place that indicate the start and end of the process. All other process nodes are on a path between them. Hence, M^{init} and M^{final} describe the process source and sink states. Moreover, a transition $t \in T$ is mathematically defined as enabled, i.e. can only be fired if

$$\forall_{p \in \bullet t}\, \sigma(p) \geq 1 \tag{2.3}$$

Hidden transitions, a special type of transition, are associated with the non-observable event ⊥. Such transitions can always fire independently of observed events as long as the introduced token requirements at incoming places are met. When firing a transition t, a token is removed from each of the input places •t, while a token is added to each of the output places t•. Process models do not always behave as desired. For example, PNs may contain unintended deadlocks or transitions that can never become enabled. Different criteria have been specified under the term soundness to prevent process models from such behavior. It is defined as follows. A labeled PN with dedicated source and sink places is considered sound iff: • for any place $p \in P$, p cannot hold multiple tokens at the same time, • for any marking $M \in M$ that indicates a token in the dedicated sink place of the PN, $M = M^{final}$ which implies that there are no remaining tokens in other places than the dedicated sink one when the final marking is reached, • for any marking M $\in$ M, the final marking M^{final} is reachable, • and for any $t \in T$, a firing sequence of events exists that enables t. Furthermore, a function $\delta_p(g)$ is introduced for all $p \in P$ that measures the average time between a token leaves a place p until a new token enters p based on an input trace c. Finally, τ_p describes the most recent time that a token entered a place p [9].

1.2.2. **Playout and Replay**

Given M^{init} of a PN, variants can be simulated by moving from one marking to another by firing transitions until M^{final} is reached. This is also known as playout. Replay is a common technique to discover deviations between recorded and modeled behavior. Therefore, each variant of an event log is replayed by executing the events sequentially on top of the PN [10].

1.3. Process Mining

Process mining is generally divided into the discovery, conformance, and enhancement of processes. Process discovery is the algorithmic extraction of process models from event logs. One can carry out analysis on obtained models which are usually in the format of PNs, Business Process Modeling Notations, Event-Driven Process Chains, or Casual Nets. In this dissertation, we will focus on PNs only. Conformance Checking is defined as the evaluation of the quality of a discovered process model, i.e. if it is a good representation of the process recorded by an event log. It is commonly evaluated based on fitness and precision among other metrics. Therefore, each trace of an event log is replayed by executing the events sequentially on top of the process model. Fitness metric functions evaluate the quality of a process model by quantifying deviations between an event log and the replay response of a process model to this event log. A process model should allow replaying the behavior seen in the event log. Precision metric functions represent the alignment between simulated traces from the obtained process model and true traces from the event log. Ideally, each generated trace by the process model should be realistic, thus being present in the actual event log. Process Enhancement considers discovered process models as well as event logs to improve or extend the models. Examples of process enhancement include structural corrections to allow the occurrence of specific behavior or extending a process model with performance data. In comparison to the introduced three main subcategories of process mining, Operational Support is a younger process mining discipline that focuses on the analysis of event data during runtime. This discipline focuses mainly on predictive objectives, such as forecasting the next event and time properties like cycle times [1].

1.3.1. **Process Discovery**

Process discovery describes the algorithmic extraction of process models such as PNs from event logs. In this section, two state-of-the-art process discovery algorithms are introduced which are used throughout this dissertation [1].

1.3.2. **Conformance Checking**

Conformance Checking describes the discipline of relating process models to event logs and the underlying system. Given a PN discovered from L that has been recorded over a finite time t, its corresponding L^+ can be related to S and PN. L^+ is usually understood as the observed realistic variants such that $L^+ \subseteq V_S$. Consequently, there might exist a subset of realistic variants that have not been observed, denoted by Vu, such that $V_S = (L^+ \cup V_u)$ and $(L^+ \cap V_u) = \emptyset$ [1].

1.3.2.1. Fitness

Fitness metric functions evaluate the quality of a process model by quantifying deviations between an event log and the replay response of a process model to this event log. A process model should allow replaying the behavior seen in the event log.

Definition 11 Log fitness is a function fit^L: $L \times PN \rightarrow [0, 1]$ that measures how much of the observed traces in L are modeled by PN. This is a quantification of the modeled and observed variants $(L+ \cap V_{PN})$ w. r. t. to $L+$.

Definition 12 System fitness is a function fit^S: $S \times PN \rightarrow [0, 1]$ that measures how much of the realistic variants is modeled by PN. This is a quantification of the modeled and realistic variants $(V_S \cap V_{PN})$ w. r. t. to V_S [1], [11].

1.3.2.2. Precision

Precision metric functions represent the alignment between simulated traces from the process model and true traces from the event log. Ideally, each generated trace by the process model should be realistic, thus being present in the actual event log [1], [12].

Definition 13 Log precision is a function prec ^{L}: L×PN →− [0, 1] that quantifies the unobserved traces that are not contained in L, but modeled by the PN. This refers to a measure that quantifies the modeled and unobserved variants ($L^{+}\{ \cap V_{PN}$) w. r. t. to L^{+}.

Definition 14 System precision is a function prec ^{S} : S×PN →− [0, 1] that quantifies how much of the unrealistic variants is modeled by PN. This refers to a measure that quantifies the modeled and unrealistic variants ($V_{PN} \cap V^{c}_{S}$) w. r. t. to V_S [13].

1.3.2.3. Complexity

Complexity is another type of metric that is used to measure the quality of the process model. It assesses how much a model is easy to understand. Several metrics are used to measure the complexity of a model. In this thesis, Control- Flow Complexity (CFC), size of the models, and structuredness measure the complexity of the models. CFC shows how many branching is prompted by the split gateways in a process model. The higher the amount of CFC value is, the more complex the model is. The size of a process model calculates the numbers of the nodes and arcs in the model. The lower the number of nodes and arcs in a process model, the lower complexity. In the end, a process model is structured. If for any split gateway in the process model, there should be a corresponding join gateway. The more structured is a process model, the simpler the model is [14].

1.3.3. Operational Support

Operational support is the youngest process mining discipline and focuses on assistance during the runtime of process cases, mostly with predictive functionalities. Typical use cases comprise the prediction of the next event, forecasting of a process' final state, or time interval prediction of future events. Predicting the next event elicits special attention since it gives organizations the ability to forecast process deviations. This type of early detection is essential for intervenability before a process enters risky states. Moreover, predictive process management assists businesses in resource planning and allocation, providing insights on the condition of a process to fulfill for instance service-level agreements [15].

1.4. Deep Learning

Deep Learning is on the uprise as continuous research with application across many domains, including process mining. For this dissertation, basic Dense Neural Networks, Generative Adversarial Networks, and Sequence Generative Adversarial Networks are of interest which is introduced in this section [16].

1.4.1. Dense Neural Networks

A neural network is a computing methodology motivated by biological nervous systems. Such networks consist of a set of artificial neurons which receive one or multiple inputs and produce one output. This set is divided into a predefined number of disjoint subsets n where $n \geq 2$. Each subset represents a layer ln in the form of a matrix containing outputs of the corresponding neurons. We refer to layer l_1 as the input and ln as the output layer of the neural network. Multiple so-called hidden layers can exist in between. In a fully connected neural network, all neurons of a layer l_k are connected to all neurons of its adjacent layer l_{k+1} for $k \leq n - 1$. A very basic neural network can be defined in the following way. A neuron j which belongs to layer l_k calculates its output based on the weighted outputs of each predecessor neuron of layer l_{k-1}. Each direct connection between two neurons i and j is associated with weight $w_{i,j}$. Each neuron j comprises a differentiable activation function ρ_j which is used to calculate the output of a neuron. Thus, the output of a neuron j belonging to l_k based on its predecessor layer l_{k-1} can be calculated as:

$$\theta_j(l_{k-1}) = \rho_j(\varphi_j(l_{k-1})) \tag{2.4}$$

It follows that:

$$\varphi_j(l_{k-1}) = \sum_i \theta_i(l_{k-2}) * w_{i,j} + w_{0,j} \tag{2.5}$$

where $w_{0,j}$ is a bias term. Such a neural network is commonly modeled as an optimization problem where a cost function ξ is to be defined as a function of the difference between neural network outputs and true values and to be minimized by adapting the weights w of the neural network. This is called a supervised learning problem [17].

1.5. Algorithms

This section introduces multiple algorithms that are used throughout this dissertation. This includes particularly certain process discovery algorithms, operational support methods [11].

1.5.1. **Split Miner**

Split miner (SM) is a process discovery algorithm that creates sound labeled PNs with dedicated source and sinks places from event logs. It is currently the best algorithm to automatically obtain PN process models from event logs with high fitness and precision. This discovery method has been developed to engage with the tradeoff between fitness, precision, and the complexity of the obtained process model. SM consists of the following five steps. First, it discovers a directly-follows dependency graph and detects short loops. In the second step, the algorithm searches for concurrency and marks the respective elements as such. Afterward, SM applies to filter such that each node is on a path from a single start node to an end node to guarantee soundness, the number of edges is minimal to reduce complexity, and that every path from start to end has the highest possible sum of frequencies to maximize fitness. Fourth, the algorithm adds split gateways to capture choice and concurrency. As the final step, this discovery method detects joins. SM encompasses two hyperparameters: a frequency threshold η to control the filtering process and ε which is a threshold to control parallelism detection. Both hyperparameters are percentiles, i.e. the numerical range is between 0 and 1. Moreover, this algorithm considers only the variants contained in a given event log [8].

1.5.2. **Next Event Prediction**

The prediction of next events is considered as a classification problem in which the probability of a next event a given the states of the process at time τ, $P(a \mid s(\tau))$, is to be found [18].

1.6. Statistical Tests

The statistical tests and methods that are introduced in this section are used throughout the dissertation to investigate the significance of proposed ideas.

1.7. Software

This section introduces relevant software that is used throughout this dissertation.

1.7.1. **ProM**

ProM is an extensible open-source Java-based framework for process mining. It was first described in 2005. ProM features a graphical user interface and allows to development and integration of plugins easily. It is therefore a de-facto standard workbench for process mining research. As of the date of this dissertation, ProM encompasses more than 1,100 official plugins [1].

1.7.2. **PM4Py**

PM4Py stands for Process Mining for Python and is a comparatively young Python project maintained by RWTH Aachen and the Fraunhofer Institute. Compared to ProM, it does not feature a graphical user interface and contains less functionality. However, PM4Py implements the most recent state-of-the-art process mining algorithms. Moreover, the platforms integrate with other Python-based data science frameworks and are therefore a useful tool when combining process mining with machine learning or deep learning [1].

1.7.3. **Tensorflow**

Tensorflow is a Python-based open-source library for differentiable programming maintained by Google. It is a de-facto standard library for machine learning and deep learning research. One of its advantages is the deployment ability on CPUs and GPUs [1].

1.7.4. **Keras**

Keras is a high-level Python-based wrapper for deep learning modules and integrates with Tensorflow. One objective of Keras is to make the development of deep neural networks user-friendly and easy [1].

1.8. Basics of Artificial Intelligence and Machine Learning

One of the definitions of AI is the study of an agent that receives data from the environment, analyzes the data, and performs an action based on the analysis. The process could be initialized by a collection of the data from the environment by sensor devices, followed by analyzing the data through machine learning algorithms, and finally, performing the action by actuators. Sensor devices and actuators are considered as the autonomous part of the AI, while machine learning techniques are the algorithmic part of the AI. In general, machine learning is considered a sub-division of AI that provides the system with the ability to learn and improve from experiences automatically. In other words, machine learning is a wide range of algorithms that build a mathematical model based on sample data or features to make predictions or decisions without being explicitly programmed to perform the task. The machine learning algorithms are capable of learning by trial and error and improving their performance over time.

1.9. Occupational Safety and Health (OSH)

The field of OSH is a subdivision of public health science and integrates disciplines such as toxicology, epidemiology, and ergonomics to study the distribution of illnesses and injuries in the workplace and implement strategies and regulations to prevent them.

IMPROVING PROCESS DISCOVERY ALGORITHMS USING EVENT CONCATENATION

This chapter describes a pre-processing approach that causes the process discovery algorithms to produce a higher quality process model, hence improving the results of evaluation common metrics. The chapter is obtained with permission of IEEE Access from a previously published work, "Improving Process Discovery Algorothms Using Event Concatenation", by the author of this dissertation [19].

1.10. Introduction

The goal of the process mining techniques is to support organizations by discovering, monitoring, and improving their processes [1]. Process mining has different applications in a variety of fields such as health care [3] insurance [4], and in the manufacturing industry [5]. Process mining consists of three steps: process discovery, conformance checking, and enhancement [6]. Process discovery transforms event logs to a process model that describes the behavior of the processes in the forms of PN, Business Process Modeling Notation (BPMN) [7], Event-driven Process Chain (BPCs), and Casual Nets (CN). Existing discovery process algorithms include Alpha-Algorithm [8], Heuristics Miner [9], Fodina [10], Evolutionary Tree Miner [11], Inductive Miner [12], and SM [13]. Conformance checking illustrates the deviations of the event logs from the process model. Finally, the enhancement step focuses on improving the process by implementing several modifications to the process model and the event logs as well. Among these three steps, process discovery plays the most fundamental step that demonstrates how process instances are executed in real life. Most process discovery algorithms use all available data in event logs to produce a process model. However, noise, inappropriate infrequent behaviors, and concurrency are presented in the real-life event logs. The expression garbage-in garbage-out underlies the fact that the poor quality of an event log would lead to poor quality of the corresponding generated process model. Therefore, the quality of the event logs information is of the utmost importance when it comes to the generation of a process model [14].

Most of the process discovery algorithms use all behavior of the data to generate a process model [15]. Although recent process discovery algorithms have been shown to remove some of the noise and infrequent behaviors in event logs [16], the resultant processed models are not clear in their execution semantics. In spite of the fact that pre-processing of the event logs is a cumbersome process, and ad hoc in its task, several methods have been developed which prioritize the motive to directly pre-process the raw data.

A general rule for filtering activities is to filter out infrequent activities from the main event logs. An example of an efficient tool that supports activity filtering is the plugin Filter Log using Simple Heuristics in the ProM process mining toolkit [17]. Another process discovery tool named Inductive Visual Miner has been discovered by Leemans et al. [12]. Inductive Visual Miner filters activities by using a slider mechanism.

Another technique in performing the pre-processing tasks is to distinguish outlier traces from event logs and to filter them out. An example is provided by Ghionna et al. [20]. This technique initially determines frequent patterns from event logs and employs a Markov Cluster processing (MCL) [21] on the traces of the event logs. In this method, outlier tracers are considered as the ones that are unassigned to a cluster and are subsequently filtered out from the event logs.

Lu et al. proposed another novel method, which uses event mapping [22] that distinguishes outlier events from the events that are part of the mainstream behavior of a process. Between two executions of the process, pairing events, which were mapped to each other, are considered similar and the unmapped pairs are considered as dissimilar in behavior. Dissimilar behaviors are considered outlier behaviors and will be filtered out from the event logs in order to generate a more accurate process model [23].

A supervised manual approach has been proposed by Cheng and Kumar, which filters noisy traces from event logs. The authors used the PRISM rule-induction algorithm [24] to train marked sub-logs, which are labeled with noisy and clean traces. Unmarked sub-logs are predicated as noisy or cleaned traces. The noisy traces are then subsequently removed from the entire logs [12].

Another related work is to build a prefix automaton of the event logs, which was recently proposed by Conforti et al. [25]. The method filters out the outlier events from the event logs by using an Integer Linear Programming (ILP) solver. A prefix automation event logs that are minimal in terms of the number of arcs is built. Infrequent arcs and subsequent events belonging to these arcs are finally removed from the event logs. Another technique has been developed by Fani Sani et al. sequential pattern mining techniques are used in this method to differentiate between outlier events and mainstream behavior events [26].

More recently, Suriadi et al. have suggested another pattern-based approach to provide a document that contains typical problems any event logs may encounter and provides solutions to these problems. The authors confirm that these document patterns can serve as a repository of knowledge for analysis that is conducted in a semi-automated manner [27].

Tax et al. have showed that filtering out chaotic activities from event logs helps to discover more accurate process models. Chaotic activities are defined to be those that can happen extemporaneously at any point in the process execution. Direct and indirect entropy-based activity filtering is used in separating chaotic activities from event logs [28].

The above pre-processing methods are useful to generate a more accurate and simpler model when an event log exhibits fewer concurrency between events. However, in cases where the event log contains many concurrent relations between events, these techniques would result in removing most of the relevant traces. Since most of the real-life processes contain concurrency between events and loops, it motivates us to discover a new pre-processing method that leads to significant results for any type of event log not just the event log with fewer concurrencies. To achieve this, we first find the probability of the frequency of pairs of events with concurrent relation in the event log. We then begin concatenating these events based on the descending order of the probability sums. In the case that two pairs of events have the same probability sums, a re-position function is performed which is explained in detail in the methodology section. In the end, self-loops are removed from the event log. The resultant event log is fed to the SM algorithm which is a process discovery algorithm to generates an efficient process model. The efficiency of the process model is evaluated by measuring common metrics such as F-Measure, and complexity on 18 real-life bench- mark datasets. We demonstrate statistical improvements in the efficiency of the process model by comparing our pre- processing approach result to that of using raw event log. Moreover, the results of our method are statistically compared to those of the 4 best recent pre-processing approaches.

1.11. APPROACH

The proposed approach is a pre-processing algorithm that aims at removing some concurrency and self-loops to improve the quality of event logs for achieving an optimal process model. The method achieves a higher F-Measure compared to maneuvers using the raw event logs. Moreover, our approach results in a much simpler and more accurate, and precise model.

The approach consists of the following steps. First, let e_i and e_j hold concurrent relations, and P stands for the probability, we find all $P(e_i||e_j)$ for all possible combinations of concurrent events, $(e_i||e_j)$ in the event logs. We introduce a threshold value of p* and only select the combinations whose probabilities pass the threshold value. Any given concurrent relation is selected if and only if both of its corresponding probabilities exceed the predefined threshold p*. We then add the probabilities of the selected combinations and sort them in descending order. We perform a re-position step for the combinations with equal probability sums. After the re-position step, we concatenate the ordered combinations on the original event logs. Finally, we remove all the self-loops that are presented on the event logs. The intuition behind concatenating events with a concurrent relation is that in many real-life event logs, concurrency accounts for a significant part of the behavior captured on the event logs. Therefore, by concatenating some of the concurrent events, the complexity of the model decreases dramatically. The various steps of the proposed model are shown in Figure 3-1.

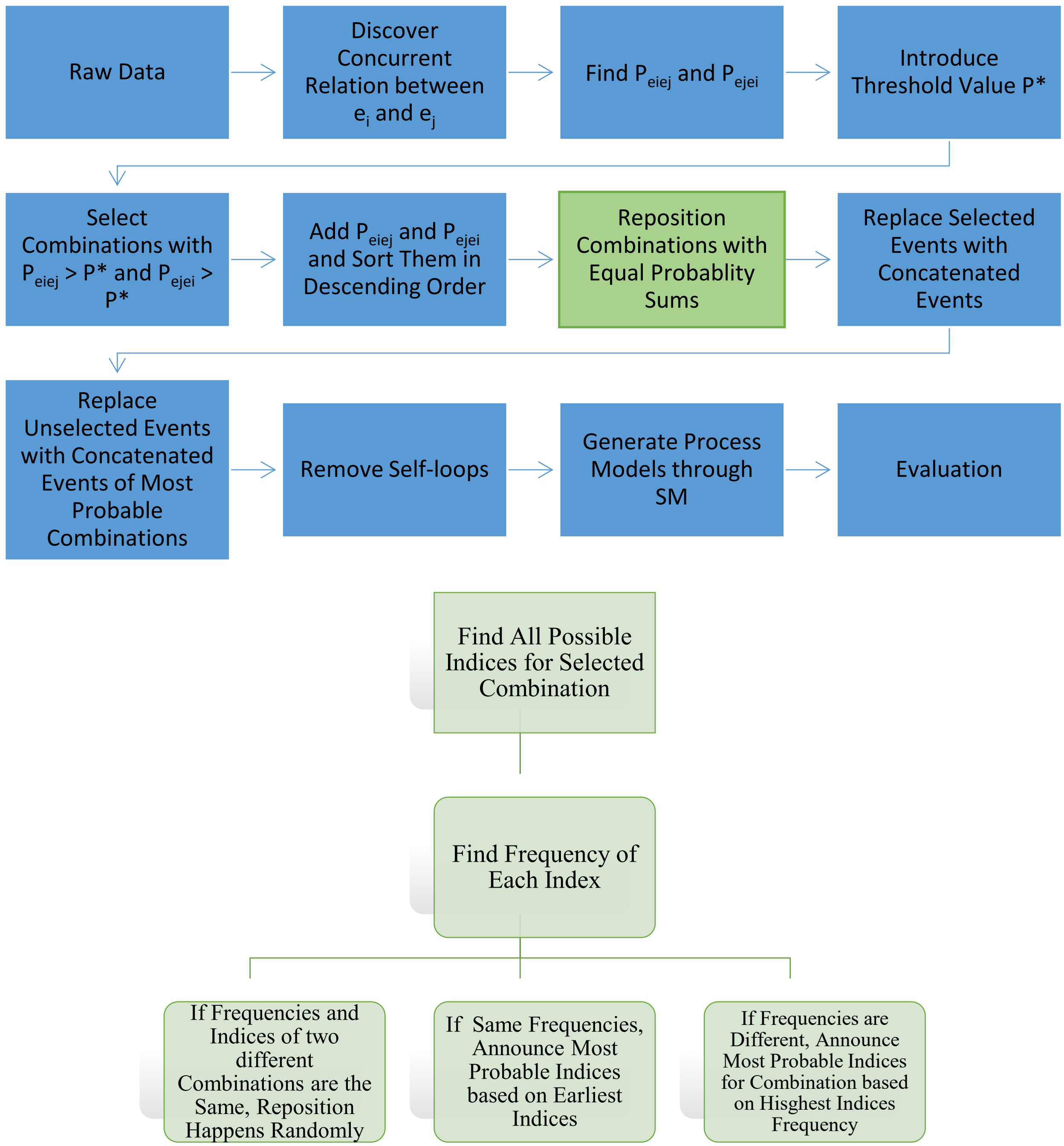

Figure 3-1: Overview of the approach

1.11.1. PROBABILITY OF THE FREQUENCY OF THE EVENTS WITH A CONCURRENT RELATION

1) Discovering concurrent events.

In this stage, we discover all concurrent relations between all sets of two events by calculating all possible combinations of two events using (n, 2) = n! /2! (n−2)! and choosing only those events that are concurrent such that ($e_i \rightarrow e_j$) and ($e_j \rightarrow e_i$).

2) Finding P($e_i \rightarrow e_j$) and P($e_j \rightarrow e_i$).

We calculate the frequency, f(e_i), for each event e_i, in each trace, $t_j \in L$. We then calculate f($e_i \rightarrow e_j$) and f($e_j \rightarrow e_i$) of only the concurrent events $e_i \parallel e_j$, and P($e_i \rightarrow e_j$) and P($e_j \rightarrow e_i$) respectively.

3) Introducing threshold value p∗.

We introduce the threshold value p∗ which denotes the cut-off probability in a given concurrent relation. The value of the threshold p∗ is found via hyperparameter optimization and selection of the highest F-Measure values.

4) Selecting combinations using p∗.

The concurrent combination of events will be chosen if and only if both P($e_j \rightarrow e_i$) and P($e_i \rightarrow e_j$) exceed the value of threshold p∗ such that:

$$P(e_i \rightarrow e_j) > p* \text{ and } P(e_j \rightarrow e_i) > p*.$$

5) Adding and sorting concurrent event probabilities.

Lastly, we take the sum of P($e_i \rightarrow e_j$) and P($e_j \rightarrow e_i$) for all concurrent events that meet the condition above. For example, let P($e_i \parallel e_j$) be the probability sum of a concurrent set of events $e_i \parallel e_j$ such that:

$$P(e_i \parallel e_j) = P(e_i \rightarrow e_j) + P(e_j \rightarrow e_i)$$

and

$$P(e_k \parallel e_l) = P(e_k \rightarrow e_l) + P(e_l \rightarrow e_k)$$

denote the probability sum of a second set of concur-rent events, (e_k, e_l). We then sort all the $P(e_j \parallel e_i)$for all concurrent events chosen in descending order the following way: $P(e_i \parallel e_j) > P(e_k \parallel e_l)$

1.11.2. RE-POSITION OF THE COMBINATION WITH EQUAL PROBABILITY SUMS

This step uses the selected combinations of step A as an input. In this stage, those selected combinations with concurrent relation are sorted, in descending order, based on their probability sums. However, we may come across several cases with equal probability sums, for which case re-position is necessary to sort them for the concatenation step. Reordering the position of those combinations is done according to the following steps.

For $P(e_i \parallel e_j) = P(e_k \parallel e_l)$

For a concurrent event (e_i, e_j) and $(e_k, e_l) \in L$ such that $P(e_i \parallel e_j) = P(e_k \parallel e_l)$.

1) For two concurrent events, $e_i \parallel e_j$ we find all the indices, $(I_s e_i, I_s{+}1\, e_j)$ of (e_i, e_j), for every trace, $t_j \in L$.

2) For $e_i \parallel e_j$, we calculate $F_s e_i e_j$ and $F_q e_i e_j$ as the total number of instances that ei and ej are concurrent at the index $(I_s e_i, I_s{+}1 e_j)$ and $(I_q e_i, I_q{+}1 e_j)$ respectively.

3) If $F_q e_i e_j > F_s e_i e_j$, then we announce indices $(I_q e_i, I_q{+}1 e_j)$ as the most probable or likely index for concurrent events $e_i \parallel e_j$.

4) We repeat steps 1−3 for all the concurrent event combinations with equal probability sums.

For $P(e_i e_j) = P(e_k e_l)$ and $F_s e_i e_j = F_q e_i e_j$

We take two concurrent events (e_i,e_j) and $(e_k,e_l) \in L$ such that $P(e_i \parallel e_j) = P(e_k \parallel e_l)$ and $F_s e_i e_j = F_q e_i e_j$.

1) Let's take two indices for the concurrent events $e_i \| e_j$, $(I_s e_i, I_s{+}1 e_j)$ and $(I_q e_i, I_q{+}1 e_j)$ such that $F_s e_i e_j = F_q e_i e_j$.

2) We find that $(I_s e_i, I_s{+}1 e_j)$ happens before $(I_q e_i, I_q{+}1 e_j)$ for $e_i \parallel e_j$.

3) We announce, $(I_s e_i, I_s{+}1 e_j)$ as the earlier indices between $F_s e_i e_j = F_q e_i e_j$

4) We repeat steps 1−3 for all the concurrent event combinations with equal probability sums and equal frequencies for the indices.

$P(e_j \| e_i) = P(e_k \| e_l)$ and $F_s e_i e_j = F_q e_i e_j$ and

$(I_s e_i, I_s{+}1 e_j) = (I_q e_i, I_q{+}1 e_j)$

In the rare scenario where two combinations may have the same probability sum and the same most likely indices in an event log, such that $P(e_j \| e_i) = P(e_k \| e_l)$, $F_s e_i e_j = F_q e_i e_j$ and $(I_s e_i, I_s{+}1 e_j) = (I_q e_i, I_q{+}1 e_j)$, we choose the order of combinations for the concatenation randomly.

1.11.3. **CONCATENATION AND REMOVING SELF-LOOPS**

1) Replacement with the concatenated event.

Thus far, the order of the combinations for concatenation is finalized in the previous step. Our algorithm selects the ordered combinations one by one for the concatenation. In order to concatenate the two events in a combination, suppose that we have a combination (e_i, e_j), we remove the event, e_i replace e_j with the concatenated event, $e_i \oplus e_j$. Note that the name of the concatenated event is derived based on the combination of 2 event names that are going to be concatenated. In this case, the name of the concatenated event will be $e_i \oplus e_j$. This step will be repeated until all selected combinations will be concatenated in an event log.

2) Replacement with the most probable combination.

After concatenating all combinations in the event log based on descending order, there will be events that are yet not concatenated. We examine each event individually that is not concatenated and see if the event, $e_i \in e_i \oplus e_j$. If that is the case, we remove e_i and replace e_i with $e_i \oplus e_j$. If e_i is not $e_i \oplus e_j$, then the final trace will remain unchanged.

3)Removing Self-Loops.

Finally, we remove all the self-loops, such that $e_i \rightarrow e_i$

1.12. EVALUATION

In this section, the proposed model is evaluated. After modifying the event logs we used the SM algorithm to generate the process models. The SM algorithm takes event logs in XES or MXML format and the thresholds of Ƞ and ε as inputs and produces a BPMN process model as an output. For evaluation, a set of publicly available logs were used. In the following parts, we provide a summary of benchmark datasets. Then we discuss the results.

1.12.1. DATASETS

4TU Center of Research Data as of August 2021 provided a collection of several real-life event logs [29]. The event logs listed in Table 3-1 were used to evaluate the proposed approach: All logs of annual Business Process Intelligence Challenge (BPIC), Road Traffic Fine Management Process (RTFMP), SEPSIS Cases log from a hospital each of which denotes the pathway through the hospital, and Hospital event log which contains information related to billing. These logs record executions of processes in a variety of fields such as healthcare, finance, and government affairs. A pre-processing step [19] was applied to remove infrequent behaviors from the BPIC15, BPIC14, and BPIC17 logs because of the complexity of the logs. Before removing infrequent behaviors, the process discovery algorithm generates a model with F- Measure close to zero due to the complexity of the event logs. The statistics of the logs are shown in Table 3-1.

Table 3-1: Statistics of the event logs

Event logs	**Original Logs**				**Pre-Processed Logs**			
	Total Traces	Distinct Traces	Total Events	Distinct Events	Total Traces	Distinct Traces	Distinct Events	% of Remaining Traces
BPIC12	13087	4366	262200	36	9291	1191	32	71
$BPIC13_{cp}$	1487	183	6660	7	1263	38	4	85
$BP1C13_{inc}$	7554	1511	65533	13	6043	111	12	80
$BPIC14_{f}$	41353	14948	369485	9	32668	9454	7	79
$BPICI51_{f}$	902	295	21656	70	884	66	70	98
$BPIC15_{2f}$	681	420	24678	82	667	417	80	98
$BPIC15_{3f}$	1369	826	43786	62	1204	810	48	88
$BP1C\ 15_{4f}$	860	451	29403	65	817	431	55	95
$BPIC17_{5f}$	975	446	30030	74	966	432	60	99
$BP1C1_{7f}$	21861	8767	7141198	41	19674	1722	12	90
RTFMP	150370	231	561470	II	103755	171	10	69
SEPSIS	1050	846	15214	16	766	425	12	73
BPIC 18-control	43808	1231	59	7	33707	41	4	77
BPIC 18-inspection	5485	6732	3190	15	4321	2197	9	79
BP1C 18-reference	43802	5463	515	6	42983	427	4	98
BPIC 19	251734	11973	9463	44	241111	10000	37	95
BP1C 20 ID	1050	846	15214	16	766	425	12	73
Hospital	100000	1020	11132	18	878	89870	12	89

1.12.2. **RESULTS**

To conduct the experiments, we used RapidProM [30] which extends RapidMiner with process mining analysis capabilities. This platform helps us to use a workflow for the experiments. The algorithm takes a threshold value of p^* to choose the combinations of concurrent events for the concatenation step. In order to find the optimal value of p^*, we ran hyper-parameter optimization which was done with the steps of 0.1. The optimal value of the p^* was chosen based on the highest value of the F- Measure. The experiment showed that the optimal value for p^* is 0.7.

Also, to choose the optimal values of SM thresholds, we used 16 different settings, i.e., the threshold Ƞ in [0.1, 0.2, 0.3, 0.4] and ε in [0.1, 0.2, 0.3, 0.4]. Note that the SM default threshold values of Ƞ and ε are 0.4 and 0.1 receptively. The optimal value the parameters were chosen based on the highest value of the F- Measure. We found that the optimal value of Ƞ and ε thresholds are 0.4, 0.4 respectively.

We used the optimal values of the thresholds for the SM algorithm, and the raw event logs to discover process models. Then, we applied our methodology to the raw event logs and used the optimal value of the thresholds for our algorithm and the SM algorithm to discover the process models from the concatenated logs. Finally, the quality of the process models was measured based on the common metrics of fitness, precision, and F-Measure as the proxies of accuracy, and size, CFC, and structuredness as the proxies of complexity. In the cases in which improvements were observed in any of the evaluation metrics after applying concatenation, the Wilcoxon Test was run to calculate the P-value and find out if the improvements are statistically significant or not. The level of significance was selected to be 0.05. The results of this evaluation are summarized in Table 3-2 and Table 3-3.

Table 3-2: Results of proposed model in terms of fitness, precision, and F Measure on event logs

Event logs	Fitness before Concatenation	Fitness after Concatenation	P-value	Precision before Concatenation	Precision after Concatenation	P-value	F-Measure before Concatenation	F-Measure after Concatenation	P-value
BPIC12	0.655	0.687	0.041	0.723	0,673	NA	0.687	0378	01142
$BPIC13_{cp}$	0.999	0.964	NA	0.931	0.974	0.039	0.964	4.986	0.041
BPIC13inc	0.998	0,951	NA	0.908	0,998	0.038	0.951	0.999	0.036
$BPIC14_{f}$	0.871	0.866	NA	0.861	0.962	0.036	0.866	0.937	0.023
$BPIC15_{1f}$	0.920	0.887	NA	0.857	0,857	NA	0.887	0,887	NA
$BPIC15_{2f}$	0.875	0.836	NA	0.800	0.804	0.054	0.836	0.835	NA
$BPIC15_{3f}$	0.782	0.785	0.053	0.789	0.748	NA	0.785	0.814	0.040
$BPIC15_{4f}$	0.815	0.827	0.053	0.839	0.839	NA	0.827	0.827	NA
$liPIC17_{5f}$	0.886	0.871	NA	0.857	0.857	NA	0.871	0.871	NA
$BPIC17_{f}$	0.956	0.876	NA	0.809	0.839	0.042	0.876	0.885	0.060
RTFMP	0.559	0.635	0.023	0.736	0.807	0.039	0.635	0.797	0.027
SEPSIS	0.976	0.659	NA	0.500	0,679	0.032	0.659	0.770	0.024
BPIC 18-control	0.823	0,845	0.041	0.779	0,810	0.047	0.810	0832	0.042
BPIC 18-inspection	0.892	0.892	NA	0.849	0.901	0.039	0.873	0.900	0.043
BPIC 18-reference	0.850	0,841	NA	0.856	0,910	0.035	0.853	0.872	0.029
BPIC 19	0.881	0.871	NA	0.812	0.889	0.028	0.851	0.881	0.041
BPIC 20 ID	0.874	0.867	0.039	66	60	0.045	260	245	0.046
Hospita1	0.921	0.915	NA	0.732	0.781	0.045	0.826	0.851	0.043

Table 3-3: Results of the proposed model in terms of complexity

Event logs	CFC before Concatenation	CFC after Concatenation	P-value	Size before Concatenation	Size after Concatenation	P-value	Structurdness before Concatenation	Structurdness after Concatenation	P-value
BPIC12	144	66	0.031	72	38	0.025	2230	2120	0.024
$BPIC13_{cp}$	16	10	0.020	12	8	0.028	68	44	0.043
BPIC13inc	19	5	0.030	14	12	0.059	80	20	0.039
$BPIC14_{f}$	33	18	0.029	21	12	0.042	304	80	0.025
$BPIC15_{1f}$	88	88	NA	72	72	NA	1320	L320	NA
$BPIC15_{2f}$	146	142	0.051	105	103	0.052	2544	3100	NA
$BPIC15_{3f}$	141	140	0.050	109	94	0.042	2578	1854	0.035
$BPIC15_{4f}$	104	101	0.050	77	75	0.052	1374	1338	0.050

Event logs	CFC before Concatenation	CFC after Concatenation	P-value	Size before Concatenation	Size after Concatenation	P-value	Structurdness before Concatenation	Structurdness after Concatenation	P-value
$BPIC17_{5f}$	111	1 1 1	NA	84	84	NA	240	240	NA
$BPIC17_{f}$	35	23	0.034	25	18	0.043	I00	74	0.042
RTFMP	50	48	0.050	32	30	0.049	5412	5410	0.053
SEPSIS	90	52	0.026	43	24	0.051	5436	4423	0.036
BPIC 18-control	145	138	0.040	61	60	0.050	982	901	0.043
BPIC 18-inspection	134	122	0.035	46	40	[1041	878	874	0,056
BPIC 18-reference	165	154	0.042	53	43	0.037	67	59	0.045
BPIC 19	155	130	0.046	46	39	0.042	404	344	0.039
BPIC 20 ID	166	131	0.039	66	60	4.045	260	245	0.646
Hospita1	176	101	0.029	67	59	0.026	409	376	0.041

The measuring tools to calculate fitness and precision, work on the PN. However, the output of SM process discovery is a BPMN model. Therefore, we needed to convert the resulting BPMN models to the PN, which is performed by using ProMs' BPMN Miner package [31]. The complexity metrics were computed on BPMN models. All the tests were performed on a computer running Windows 10 with an Intel i7-6700 CPU and 16 GB RAM.

Table 3-2 shows the experimental results for the best obtained F-Measure values and corresponding fitness and precision values before and after preprocessing. Our results show that the F-Measure values were significantly improved by concatenating the events, which have a concurrent relation in the real-life event logs. 14 of 18 benchmark datasets have clearly shown major improvement in F-Measure values.

Table 3-3 shows the results of the comparison of the discovered process models with respect to their complexity both before and after the preprocessing. The complexity of the models has been evaluated based on three different metrics; CFC, size, and structuredness, which we have mentioned earlier in the previous section. All of these metrics are inversely related to the understandability of the process models. Based on our results, 16 out of 18 benchmark datasets clearly showed that the process models that are generated through SM after the pre-processing step are simpler in terms of CFC. Besides, the overall size of the models is also found to be greatly reduced as shown in 17 out of 18 benchmark sets. Moreover, the structuredness of the models have improved in 14 out of 18 benchmark datasets. These results point out that our models are much more feasible to understand and effective in performance with less complexity.

Based on Table 3-1, the improvements in F-Measure and complexity resulted from decreasing/removing the infrequent traces which cause minimal scarification in fitness but increase much in the precision value. In some event logs, our approach removed as much as 30% of the infrequent traces from the original event logs in order to reach the best value of the F-Measure. The percentage of the remaining traces in the event logs after pre-processing is shown in Table 3-1.

Another reason which leads to the improvements of the F-Measure values and the complexity of the process models by applying our approach is that concatenating some concurrent relations, removes some of the unnecessary behaviors from the event logs. As seen in Table 3-1 the numbers of distinct events after pre-processing steps are decreased which indicates that unnecessary behaviors are removed and caused significant improvements in F-Measure values and the complexity of the process models.

RTMF dataset contains high frequent behavior, whereas the SIPSIS dataset contains many distinct traces that happen only once. Also, both these datasets contain many concurrencies and loops compared to other datasets. The algorithm was able to successfully eliminate concurrency of data and remove self-loops by effectively concatenating a large set of events that have concurrency relations with each other. As a result, most infrequent behaviors, and infrequent traces in the event logs were removed. Therefore, the algorithm was able to facilitate the process discovery algorithm to generate a process model that has distinctly higher F-Measure values and lower complexity. To elucidate further, most of the BPIC15 datasets contain only fewer numbers of concurrency relations between events compared to other datasets. Even when the concurrency relation exists, the probability value of the first event follows the second event that exceeds the p^* threshold value of our algorithm. However, the probability of the second event following the first event does not pass our p^* threshold value. As a result, our algorithm did not choose these combinations for concatenation. In fact, our algorithm selects combinations, which are having strong concurrent relations for concatenation. As a result, the F-Measure values for BPIC15 datasets either have improved slightly or have not changed compared to the results of using just the raw data. In terms of complexity, even though our algorithm made minor changes in these event logs, still the complexity of the process models has improved in most of the BPIC15 datasets.

Next Event Prediction for Critical Health Outcomes

This chapter describes a process mining/ deep learning approach to predict three different healthcare outcomes. More specifically Chapter 4.1 predict unplanned 30-day readmission of ICU patients with Heart Failure. Chapter 4.2 describes similar approach to predict mortality of the COVID patients every 6-hours within 72 hours of their admission. These two sections are obtained with permission of BMC journal from the previously published works, “Prediction of Unplanned 30-day Readmission for ICU Patients with Heart Failure " and “Process Mining- Deep Learning Approach to Predict Survival Outcome for COVID-19 Patients”, by the author of this dissertation [32], [33]. Chapter 4.3 uses process mining/ deep learning approach to predict mortality in Paralytic Ileus after 24 hours of the patient being admitted. The chapter is published in 2021 International Conference on Cyber-Physical Social Intelligence (ICCSI) [34]. The details of each case studies and their results are explained below.

4.1 PREDICTION OF UNPLANNED 30-DAY READMISSION FOR ICU PATIENTS WITH HEART FAILURE

This chapter describes a process mining/ deep learning approach to predict unplanned 30-day readmission of ICU patients with Heart Failure. The chapter is published in BMC journal [32].

4.1.1 INTRODUCTION

The prevalence of Heart Failure (HF) rises over time. Approximately, 6 million American adults (age > 20) had HF between 2015 to 2018 [35]. Despite the progress made in HF therapeutics promising to improve mortality, readmission rates remain high at nearly 20% [35]–[37] Excessive unplanned readmissions and subsequent waste of medical resources have had financial implications that directly affect the overall performance of the hospitals. The Hospital Readmissions Reduction Program was established by the Affordable Care Act (ACA) in 2010 to encourage hospitals to avoid readmissions by penalizing the hospitals that exceed the expected thresholds [38]. Since 2012, hospitals have been penalized over $2.5 billion by the Centers for Medicare & Medicaid Services (CMS) for exceeding the unplanned 30-day readmission rates [39], [40]. Unplanned ICU readmission may impose a severe financial burden on both hospitals and patients [39]. Readmissions were found out to be associated with increased morbidity and mortality. The mortality rate of unplanned ICU readmission ranged between 26% to 58% [41]. The ICU readmission rate had increased over time rising from 4.6% in 1989 to 6.4% in 2003 [35]. Approximately 16% of ICU unplanned readmission occurred within 30-days of initial hospital discharge [42], [43].

The EHR has been revolutionizing the health care decision-making processes through collecting and preserving medical data in a digital format. The use of the EHR has been allowing hospital systems to make intelligent data-informed decisions to address a wide range of problems from learning personalized prescriptions to maximizing the performance of hospitals [39].

Several machine learning and AI techniques have been proposed to predict unplanned 30-day readmission of ICU patients with HF [42], [44]. However, the results of the developed models were not quite reliable. Process mining analyzes and optimizes the sequence of events occurring during running processes, known as the event logs. The process mining approach has been used to enhance the healthcare processes [45]. However, there has been a scarcity of considering the medical history of the patients from past hospital visits for readmission prediction [34], [46], [47].

The present study aimed to introduce a novel process mining approach that incorporated the past medical history of the patients from prior hospital visits and the time information related to the variables (Time State Samples (TSS)) to predict unplanned 30-day readmissions of HF patients.

4.1.2 APPROACH

4.1.2.1 DATA SOURCE AND INCLUSION CRITERIA

We used the Medical Information Mart for Intensive Care III (MIMIC- III) public database, which contained deidentified clinical data of the patients who were admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts [48]. MIMIC- III contained 38,597 adult patients and 49,785 hospital admissions from 2001 to 2012. This database consisted of various tables such as admission information, demographics, caregiver information, lab values, charted observations, discharge summary notes, and diagnosis codes.

In order to identify HF patients, specific ICD-9 codes related to the HF diagnosis including 398.91, 402.01, 402.11, 402.91, 404.01, 404.03, 404.11, 404.13, 404.91, 404.93, 428. XX were used in this study. Patients were included if any of these ICD9 codes appear in the most recent admission following the standard approach in the existing literature [45]. We also included HF patients who had visited any hospital at least once before the current ICU hospital visit. Patients who died in the same hospital ICU or got discharged from a hospital ICU and died later in another hospital or other parts of the same hospital were excluded.

4.1.2.2 VARIABLE SELECTION

Several variables were considered as the inputs to the model which are as follows: the admission type, the associated time of the admission, types of insurance, the discharge time, several lab measurements, various performed services, procedures, and diagnoses on the patients, and demographic information. The admission types were categorized either as planned or unplanned admission. The insurance group types were defined as Medicare, Medicaid, Private, Government, and Self-pay. Lab values typically obtained to predict HF were extracted for each patient including Blood Urea Nitrogen (BUN), Serum Creatinine, sodium ion, and pro-brain natriuretic peptide (NT-proBNP). The various types of performed services, procedures and diagnoses were considered in the form of CPT and ICD-9-CM codes. The demographic information including age, gender, and ethnicity was used as additional variables in the model.

4.1.2.3 CONVERSION OF EHR INTO EVENT LOGS

The proper format of the input data for the process mining is event logs. Event logs contain the sequences of events as well as the associated time at which specific events occurred, which was referred to as timestamps. The transformation of the EHR of the patients into the event logs was done based on the method reported by Theis et al. (2021) [47].

Thirteen different event types were defined in this section. shows the mapping of each of the considered event types with the MIMIC-III tables.

Table 4-1: Mapping of the medical health records of the patients from the MIMIC- III database to the event logs

MIMIC-III Tables	Events
ADMISSIONS PATIENTS	Admission type event Insurance type event Discharge event
ADMISSIONS LABEVENTS D- LABITEMS	BUN Mean event Serum Creatinine Mean event NT_ proBNP Mean event Sodium-ion Mean event BUN std event Serum Creatinine std event NT_ proBNP std event Sodium-ion std event
ADMISSIONS DIAGNOSES_ICD	Elixhauser Comorbidity Score events
DIAGNOSES_ICD PROCEDURES_ICD CPTEVENTS	30 Artificial event abstractions

BUN, Blood Urea Nitrogen; Pro-BNP, Pro-brain natriuretic peptide; std, Standard Deviation

4.1.2.4 EVENT TYPES AND ASSOCIATED TIMESTAMPS

For each patient, we converted the EHR to events with the following sequence:

First, we considered the admission type event with the admission time as its timestamp. Admission type event was important since it distinguished whether the admission was planned or unplanned. The second event was the insurance type event with a timestamp of 1ms after the timestamp of the admission type to maintain the order of the events. Insurance type was chosen since it could possibly affect the discharge/transfer rate.

The next set of events were the lab measurements. Specific HF-related lab measurements were chosen based on the literature [45] and experts' opinions. The lab measurements might be measured once or several times for each patient. Two types of events were created for each lab item, out of which one was the Mean of the specific lab item and the other was the std of each lab item.

In the cases in which the lab item was measured once, the timestamp of the Mean event was set to the timestamp at which the lab item was measured initially. The timestamp for the std event was set to 1ms after the timestamp of the Mean event. For the lab measurements, which were measured several times, we performed similarly for the Mean event. However, for the std event of such cases, the timestamp was set to the last time at which the lab item was measured.

We considered a separate set of events representing the comorbidities. Elixhauser comorbidity score was calculated by using the ICD-9 diagnosis codes for each hospital visit [49]. A specific comorbidity group can be determined by assigning points through the Elixhauser comorbidity score if particular ICD-9 codes are present. These events were created since they represented how critical a patient was, if a patient had chronic diseases or not, and what diseases had been diagnosed over time. Additionally, based on the literature, these events were strongly associated with ICU readmission risk [50]. In cases where a point was assigned to a specific group, an event was created with the same name as in the comorbidity group. Since the focus of this work was on readmission prediction, we needed specific event logs that separated all the timestamps of the events from the discharge timestamp (which was the final event). Therefore, the timestamp for these events was set to be very close to the discharge time of the relevant hospital encounter. In cases where multiple comorbidity events were created, the timestamps for the second comorbidity event were set 1ms after the timestamp of the first event. The same logic was applied for the next comorbidity events as well.

The artificial events were created from the sequence of CPT, and ICD-9-CM observations codes as by Theis et al. [47]. We considered these events since they represented the diagnoses and procedures of a patient which were likely to be important factors to predict readmission. The artificial events' timestamps were set to the timestamps of the sequence of the observations plus 1ms. These timestamps accordingly were compared with timestamps of the discharge event, and they were set to a time before the discharge timestamp to ensure the orders of the events were maintained.

In the end, the discharge event was created for each hospital admission and the timestamp of this event was set to the discharge time of the patient for the corresponding admission. This event was created since this was a point at which the next event (unplanned 30- day readmission) would be predicted by using DREAM algorithm [2].

Note that the addition of 1ms to the timestamps in our conversion did not alter any information since the time dimension in MIMIC- III was days and subsequently negligible in our analysis.

4.1.2.5 UNPLANNED 30 READMISSION PREDICTION

We proposed a process mining approach for unplanned 30-day readmission prediction. The resultant event logs were fed to the DREAM algorithm to generate the time information (TSS). The severity scores on admission day including the Charlson [51] and Elixhauser scores were used as independent variables. Charlson score method assigns higher weights to more severe and critical conditions as compared to Elixhauser that assigns the same weight to all conditions. The generated TSS, together with the demographic information and the severity scores were then fed to a NN model to predict unplanned 30-day readmission of the ICU patients with HF. Figure 4-1 illustrates the overview of the proposed model.

The proposed model was evaluated by calculating AUROC, precision, sensitivity, and F-score on the test set. To obtain 95% CIs of the AUROC value, DeLong's method was used [52].

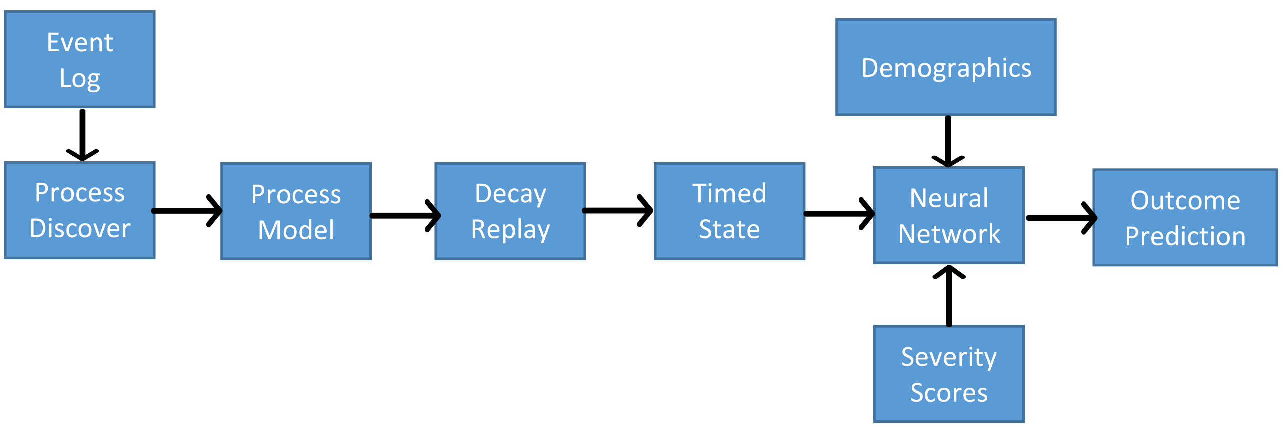

Figure 4-1: Overview of the methodology

4.1.2.6 STATISTICAL ANALYSIS BETWEEN COHORTS

The derivation and validation cohorts were compared using Chi-Square and two-sided t-tests. For the comparison of the categorical variables, Chi-Square tests were performed, and for continuous variables, t-tests were implied. The significant level was determined based on $P < 0.05$. Descriptive statistics, model development, and statistical analysis were conducted using Python, version 3.6.

4.1.2.7 VARIABLES IMPACT

Shapley value analysis [53] was conducted on the test set to find out the impact of each variable in our model prediction and to figure out which variable was particularly associated with readmissions. The Shapley values described the Mean contribution of each variable to the outcome across different coalitions [47].

4.1.3 RESULTS

4.1.3.1 COHORTS CHARACTERISTICS

Following the approach for selection of HF patients discussed before in this work, a subset of 3411 patients was selected from the MIMIC-III database. The selected cohort was then split into derivation and test sets randomly with a ratio of 84/16, which yielded a result of 2856 patients for derivation and 555 patients for the test cohorts. Moreover, the derivation cohort was further split into derivation and validation with the ratio of 85/15 resulting in 2422 patients for derivation and the remaining 434 patients for validation cohorts, to be used in the process discovery step and the NN derivation. Lastly, the best model was chosen for further evaluation on the test cohort. The description of the derivation and validation cohorts is presented in Table 5. In terms of age, the validation cohort (70.4 years) was slightly older than the derivation cohort (69.9 years) with a P of 0.228 which showed there were no significant differences between cohorts. In terms of gender, the derivation cohort (47.6 %) contained slightly more females compared to the validation cohort (46.3 %). The whole distribution of the race was not significantly different between the cohorts [P = 0.270], of which the details are shown in Table 4-1. The proportions of white patients in the derivation and validation cohorts were 75.8 %, and 74.9 % respectively. The lab measurements were not significantly different between cohorts except for Urea Nitrogen which was 0.017.

Table 4-2: Comparison of the variables including outcome, demographics, and laboratory findings between derivation and validation cohorts

Characteristics	Derivation Cohort (N = 2422)	Validation Cohort (N= 434)	P
Outcome Variable N, (%)			
Readmission	581 (23.9)	102 (23.5)	0.270
Demographics			
Age Mean (std)	69.9 (14.3)	70.4 (13.9)	0.228
Female (%)	47.6	46.3	1.00
Race N (%)			0.270
African American	291 (12.0)	51 (11.8)	
Hispanic	76 (3.10)	18 (4.15)	
Others, non- Hispanic	170 (7.00)	37 (8.53)	
White	1835 (75.8)	325 (74.9)	
Asian	50 (2.10)	3 (0.691)	
Laboratory Findings Mean (std)			
Sodium	138.6 (4.58)	138.5 (4.80)	0.835
Urea Nitrogen	34.4 (24.0)	32.6 (22.3)	0.017
NT proBNP	0.187 (0.380)	0.181 (0.385)	0.613
Serum Creatinine	0.002 (0.04955)	0.004 (0.062)	0.083

Pro-BNP, Pro-brain natriuretic peptide

4.1.3.2 Shapley Value Analysis

Figure 4-2 illustrates the results of the Shapley value analysis. Based on this figure, severity scores had the most significant impact on the prediction of unplanned 30-day readmission of the HF ICU patients, followed by demographic information and admission events that seemed to have a similar impact on prediction. Whereas artificial events, comorbidity events, and lab measurement events were the least important variables for the prediction of the outcome in order. Among the severity scores, Charlson had a higher impact on prediction as compared to that of Elixhauser which showed that the severity of the conditions played an important role in the prediction of unplanned 30-day readmission of ICU HF patients since Charlson score assigns higher weights to the severity level of the conditions than Elixhauser.

The Shapley value analysis confirmed that the severity scores had the highest impact on prediction in our model. However, there were other contributing factors including demographics, admission events, artificial events, comorbidity events, and lab measurement events impacting the prediction of the outcome which were all ignored by health calculators as the inputs for prediction.

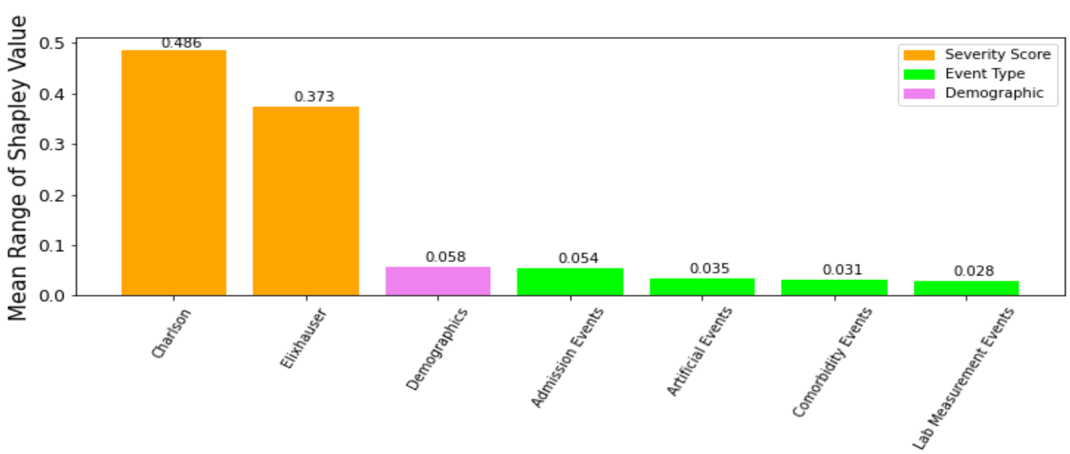

Figure 4-2: The Mean range of Shapley Values for each variable type

4.1.3.3 Evaluation Metrics and Proposed Model Performance

The proposed approach resulted in the following metrics, AUROC, 95% CI, precision, sensitivity, and F-score are as follows, respectively: 0.920, CI of [0.891- 0.962], 0.880, 0.781, 0.800.

4.1.3.4 DISCUSSION

4.1.3.5 EXISTING MODEL COMPLETION SUMMARY

Several methods have been concurrently developed to predict unplanned 30- day readmission of the ICU patients with HF aiming to benefit both health care providers and the patients. Table 4-3 shows the existing models which have been developed to predict unplanned 30-day readmission of ICU patients with HF by using the MIMIC- III dataset. As a result, the same outcome, dataset, and patients were used compared to that of ours.

Table 4-3: Summary of the existing models and their performance on the MIMIC-III dataset

Study	Method	Variables	Performance
Lin et al. [44]	Recurrent Neural Networks (RNN), RF, CNN	GCS eye, GCS verbal response, GCS motor response, Capillary refill rate, DBP, SBP, MBP, HR, Glucose, FIO, OS, RR, Body Temperature, pH, Weight, Height, Gender, Age, Insurance type, Race ICD-9 embedding	AUROC of 0.791 95% CI: 0.782–0.800
Hu et al. [42]	constrained Support Vector Machine (cSVM)	BUN, DBP, FIO, Glucose, HR, RR, SBP, Temperature, Weight, pH, FIO, HR, MBP, OS, RR, SBP, Temperature, Weight, LOS, GCS eye, GCS verbal, Age, Gender, Race, Insurance, Discharge location	AUROC 0.680 95% CI: 0.651–0.722
Baruah [46]	CNN	Clinical notes	AUROC 0.646 Precision 0.876 Sensitivity 0.697
Liu et al. [54]	Random Forest (RF), Convolutional Neural, Networks (CNN)	Clinical notes	Precision 0.698 Sensitivity 0.771 Accuracy 0.733
Huang et al. [55]	bidirectional transformer model (Clinical Bert)	Clinical notes	AUROC 0.768

BUN, Blood Urea Nitrogen; DBP, Diastolic Blood Pressure; FIO, Fraction of Inspired Oxygen; HR, Heart Rate; RR, Respiratory Rate; SBP, Systolic blood pressure; MBP, Mean blood pressure; OS, Oxygen saturation; LOS, Length of Stay; GCS eye, Glasgow coma scale eye opening; GCS verbal, Glasgow coma scale verbal response. aMean and std were used as continuous variables in Lin et al. [44] and Hu et al. [42].

In this study, a process mining technique was investigated for predicting unplanned 30-day readmission of ICU patients with HF, in which time information associated with the events, severity scores, and demographics were fed into a NN model.

The effectiveness of our developed approach outperformed the best results of the existing literature in terms of the AUROC value proposed by Lin et al [44]. The efficacy of our approach was demonstrated by a substantial improvement of +10% on AUROC.

In addition, the presented results indicated +6% and +7% improvements in sensitivity and F-score metrics, respectively, compared to the best sensitivity and F-score values reported in the literature by Liu et al [54].

Although the existing proposed methodologies in the literature were successful in predicting unplanned 30-day readmission of ICU patients with HF, they possessed several drawbacks. First, most of the existing models did not use the time-series features, and to the best of our knowledge, none of them incorporated time information associated with the variables in their predictive modeling that could lead to significant information loss and poor performance accordingly [55].

Furthermore, the proposed approach was a process mining approach that illustrated the careflows of patients through a process model. As a result, our framework was more interpretable compared to the existing methods, which is significant for clinical applications [56].

Moreover, the health calculators that computed outcomes based on the severity scores ignored the past medical history of the patients which could have a significant impact on the likelihood of unplanned readmission.

Our proposed approach had several advantages over prior research papers which are as follows: a) Process mining approach yielded a comprehensive analysis of careflows of patients through a process model which was understandable and could easily be interpreted compared to machine learning techniques. The process model provided a map that represented the possible diagnosis, procedures, performed services, laboratory measurements, and more, that could happen to a patient. Additionally, it eased the interpretability of a model prediction. An example of a process model can be found in the existing research paper [57] b) The EHR can be directly used as inputs to our proposed approach without any computationally expensive preprocessing steps. c) The process mining framework was capable of modeling the time-related variables and incorporating the medical history of the patients from the previous hospital visits in the prediction algorithm unlike machine learning based models and health calculators.

4.1.3.6 Study Limitations

The proposed approach had some limitations. Since this approach was a process mining approach, the availability of the past hospital visits of the patients was essential. This approach was not useful for patients whose admission histories were not available. However, this limitation can be overcome if the history of patients could be exchanged through a network system between health care providers. Application Program Interfaces (APIs) and similar innovations hold promise that soon these drawbacks can seemingly be curtailed.

Moreover, in our model development, the derivation and validation datasets were used to build the model. The test dataset was set aside from the beginning and only used to evaluate the performance of the model. The derivation, validation, and test sets were coming from the MIMIC-III dataset. However, using an independent dataset from a different system would be beneficial to test the performance of the model [3], which provides room for future work.

4.2 PROCESS MINING- DEEP LEARNING APPROACH TO PREDICT SURVIVAL OUTCOME FOR COVID-19 PATIENTS

This chapter describes a process mining/ deep learning approach to predict mortality of the COVID patients every 6 hours within 72 hours of their admission. The chapter is published in BMC journal [33].

4.2.1 INTRODUCTION

With the rapid emergence of the COVID-19 pandemic, the use of machine learning and AI to understand and predict. virus spread, potential vaccines, morbidity, mortality, and resource allocation have become paramount. The modeling of morbidity and mortality has yielded great insight into how individuals with COVID-19 progress through the illness [58]. These advances have informed hospitals and administrators on the types of care that have proven effective in other settings. As the COVID-19 pandemic continues, the virus has mutated, care plans have changed, thus the use of static modeling is likely an ineffective tool for understanding how to provide care from both the patient and public health perspective [59].

Process mining techniques assist in analyzing and optimizing systems using sequences of observations. Process mining approaches have been shown to be a valuable tool set in the healthcare industry. One of the applications of process mining in healthcare is to enhance healthcare processes [45]. However, it has not yet been used to predict mortality after admission of the COVID-19 patients to the hospital.

The present study aimed to develop a process mining/deep learning approach to predict mortality among COVID-19 patients. The process mining incorporated typical variables used in prior prediction models [34], [47], but in addition, used time information related to the variables, TSS. We were interested in a model which was able to update the prediction in the 6-hour intervals within the first 72 hours after admission to the hospital. In addition, to the process mining approach, published and self-developed traditional machine learning models were utilized as baselines which did not use time as a variable. We postulated that since traditional models lack direct use of time information they will not perform as well as the process mining approach during the interval evaluation.

4.2.2 APPROACH

4.2.2.1 DATA SOURCE AND VARIABLES

UIH is a tertiary, academic teaching hospital in Chicago. The University of Illinois at Chicago (UIC) Institutional Review Board approved this study. All admissions to UIH for COVID-19 positive patients were reviewed for the time of the first COVID-19 positive test and the date of admission. If the first positive COVID-19 test was performed greater than 14 days prior to admission or greater than 48 hours after admission, the patient was excluded. Patients transferred from another institution were reviewed for prior COVID-19 testing. The patient was excluded if the most recent COVID-19 test has been performed longer than 14 days prior to the transfer. If the transfer was not related to any possible COVID-19 symptoms, the patient was excluded. Symptomatic patients for COVID-19 were included in this cohort, as verified by manual chart review or claim data.

If a patient had multiple hospital admissions at UIH related to COVID-19, each admission encounter was flagged as death or discharge. Additionally, admissions were flagged as ICU or Non-ICU. This resulted in 508 COVID-19 admissions of 481 unique patients, 11.8% of those admissions resulted in death, 36.4% led to at least one ICU transfer, and 11.2% of the admissions led to an ICU transfer with death outcome.

We partitioned our data randomly into training, validation, and test cohorts using a 60/20/20 split ratio, respectively. Consequently, each admission encounter belonged to a unique cohort.

Variable selection was based on a review of the extant literature and expert opinions as per prior work which are demographics, vital signs, laboratory data, and clinical characteristics (comorbidities, diagnosis codes, problem list, clinic notes, procedure reports, location within the hospital) were assessed.

4.2.2.2 CONVERSION OF EHR INTO EVENT LOGS

In process mining, events consist of a name that describes the observed action and the corresponding timestamp of occurrence, i.e., when the event has been observed. The temporally ordered sequence of such events is called a trace. Commonly, a trace contains only events that belong to the same context. In this research, the observations of specific COVID-19 admission of a given patient formed a trace. This can also be understood as a trajectory. The set of all traces, i.e., all COVID-19 admissions in the dataset, comprised an event log.

The extracted traces of the event log, and consequently the predictions to be performed, were based on COVID-19 admissions. Predictions were performed after 6h, 12h, 18h, 24h, 30h, 36h, 42h, 48h, 54h, 60h, 66h, and 72h of the hospital admission. Obviously, patients that had been discharged or patients that died before a given time of prediction were excluded. Hence, predictions were only performed for patients that are at the prediction time present in the hospital and had not yet been discharged or died.

For each admission, static features were extracted that did not change over the course of the hospital encounter (i.e., demographic information, comorbidities). The patient-centric trajectory of the hospital encounter was then represented as a trace. A trace started with the first occurrence of an event related to the hospital encounter and ended with the occurrence of an outcome event, in the case of this work, either discharge or death. Each event was associated with the timestamp of observation. In this way, the state of the patient can be reconstructed at each point of time. Events can be either location-based, vitals, lab measurements, report-based, encounter-based, or ICU-based.

Location-based events represented that a patient moved to a particular location. For example: the emergency room, ICU, non-ICU inpatient teams, among others. Vital sign events represented the observation of a particular vital sign, which were subsequently recorded as either "ok" or "critical". Laboratory measurements were flagged as either normal or abnormal to create the laboratory events. Report-based events corresponded to procedure reports (e.g., electrocardiograms or radiological testing). Report-based events correspond to a performed procedure without considering individual findings or outcomes within the reports. Encounter-based events represented specific highlights (admission, observation status, discharge, or death) during the hospital stay. ICU-based events were based on the admission or not to the ICU at UIH, therefore, there were ICU-in, and ICU-out events recorded.

After the conversion of the EHR data, a set of traces, i.e., an event log, was obtained. Each trace corresponded to one hospital admission and used the events to describe the health trajectory of the patient from admission to either discharge or death. Due to the definition of events and the sequential structure of traces, the traces could be used to create sub-races, such that a sub-trace contained only events from, e.g., admission time to 24h into the hospital encounter.

4.2.2.3 MODEL DEVELOPMENT FOR PREDICTION

A process mining/deep learning model was proposed to predict the mortality likelihood of a given COVID-19 patient every 6-hour within the first 72 hours after the patient was admitted to the hospital. The patient trajectories were used to extract a process graph model using a process mining discovery algorithm [9]. The resulting process model and the patient trajectories from admission to the time of prediction were fed to the DREAM algorithm [10]. The DREAM algorithm enhances the process model with functions that parameterize time using the patient trajectories. As an output, the DREAM algorithm provides a state of the process model for each patient that contains time information. Hence, the output of the DREAM algorithm is called TSS. The TSS corresponds to the health condition of a patient up to the time of prediction and contains information of the observed events and process states, and their interarrival times. The comorbidities and the demographic information were used as independent variables. The generated TSS, together with the demographic information and the comorbidities were then fed to a NN model to predict mortality of the patients for each 6-hour interval within the first 72 hours. Note that for all time intervals, the same process model was used, and the architecture of the NN was changed automatically based on the new information which was fed to the NN every 6-hour within the first 72 hours after the patient was admitted to the hospital. Figure 4-3 illustrates the overview of the complete model. Descriptive statistics, model development, and statistical analysis were conducted using Python, version 3.6.

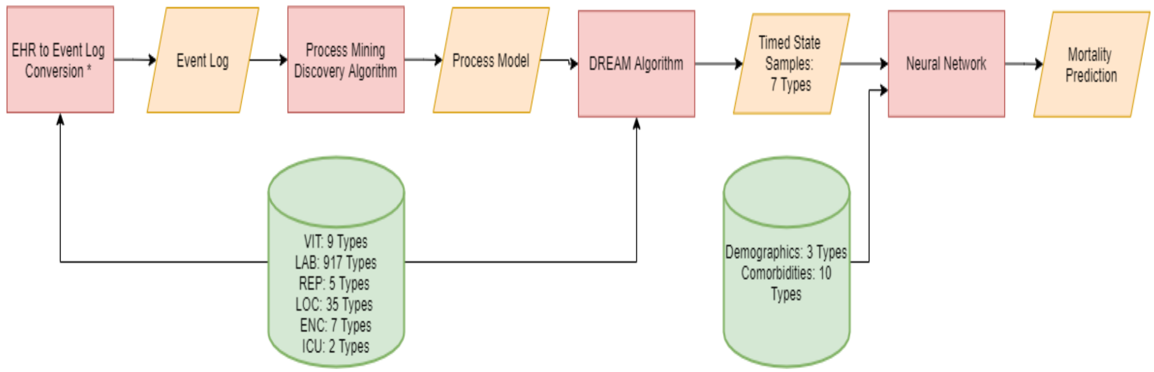

Figure 4-3: Process mining/deep learning Model Development.

Orange parallelograms represent input/ output data. Four algorithms used in the methodology are shown in red rectangles. Green cylinders represent variable types coming directly from the database that were used as the inputs to the algorithms.

4.2.2.4 BASELINE MODELS

To compare the results of the process mining approach, a published model and self-developed model were utilized. Both models were developed using machine learning algorithms that were not directly utilizing time information [33]. We used a Logistic Regression (LR) model [60] of 305 patients from China [61]. Core features in this model were age, Lactate dehydrogenase (LDH), and C-reactive protein (CRP).

The self-developed model was trained based on the UI Health data cohorts to explore other machine learning algorithms for the time interval modeling task. The development of these models utilized the same possible information as presented in the UIC Cohort and Variables subsection. However, the data was kept in the original tabular format, as opposed to the event log format. The time component of the data was implicitly added to the training process by splitting a single training instance into multiple instances based on the time interval. This conversion allowed the developed models to witness instances from low time intervals that had limited information and from high intervals with more complete information. A variety of popular machine learning algorithms were evaluated to classify mortality at each 6- hour time interval within 72 hours of admission. These algorithms included LR [60], Decision Trees [62], Random Forest [63], XGBoost [64], LightGBM [65], and CatBoost [66]. The training process of these models included both a forward step feature selection and a grid search of model parameters. This search process aimed to find the best model with the least amount of input features. The best model was determined based on the AUROC of the validation cohort at each time interval.

4.2.2.5 MODEL EVALUATION

The primary evaluation metric for model development and selection was the AUROC. We used Delong's test to calculate 95% CI of the AUROCs and compare AUROC CIs between models [52]. In addition, we calculated the sensitivity and specificity of models across the time intervals [52], with 95% CIs.

4.2.2.6 ANALYSIS OF CONTRIBUTION OF PROCESS MINING UNIQUE VARIABLES

Shapley value analysis [53] was conducted on the testing cohort to find out the impact of each variable in the process mining model prediction and to identify variables associated with the mortality prediction of the COVID-19 patients in the 6-hour intervals within the first 72 hours, and to compare it to the machine learning and LR models.

4.2.3 RESULTS

4.2.3.1 UIH COHORTS CHARACTERISTICS

Table 4-4 shows the demographics, clinical characteristics, and medical conditions of the study sample per encounter. There was a total of 508 encounters of 481 unique patients. The training cohort included 303 encounters (60%), the validation and testing cohorts the remaining 101 (20%) and 104 (20%) encounters, respectively. The testing cohort was slightly younger, 53.4 vs 56.6 (training and validation) years old [P = 0.178]. Though the whole racial distribution was not significantly different between the cohorts, the proportion of self-declared black patients was slightly higher in the validation and testing cohort compared to the training cohort.

Table 4-4: Encounter characteristics of the training, validation, and testing cohorts

Characteristics	**Training cohort (N = 303)**	**Validation cohort (N=101)**	**Testing cohort (N=104)**	**p-value of Statistical Test on Train vs. Test**		**p-value of Statistical Test on Validation vs. Test**	**p-value of Statistical Test on Train + Validation vs. Test**
Number of distinct Patients N (%)	288 (95.0)	96 (95.0)	97 (93.3)				
Primary Outcome (N, %)							
Mortality	43 (14.2 %)	6 (5.9 %)	11 (10.6 %)	0.175		0.115	< 0.0001
Demographics							
Age in years Mean (std) Patients older than 89 have been clipped to age 90	56.6 (16.6)	56.6 (15.6)	53.4 (14.2)	0.012		0.028	0.009
Female N (%)	147 (48.5)	50 (49.5)	56 (53.8)	0.175		0.268	0.178
Race/ethnicity (N, %)				0.632		0.946	0.755
African American	137 (45.2 %)	51 (50.5 %)	49 (47.1 %)				
Hispanic	36 (11.9 %)	13 (12.9 %)	16 (15.4 %)				
Others	112 (37.0 %)	30 (29.7 %)	32 (30.7 %)				
White	18 (5.9 %)	7 (6.9 %)	7 (6.7 %)				
Mean (std) of the number of laboratory measurements per encounter							
	636 (786)	510 (663)	531 (972)	0.078		0.228	0.090
Mean (std) of the number of vitals sign measurements per encounter							
	999 (1540)	765 (1344)	802 (1971)	0.026		0.124	0.030
Comorbidities				0.816		0.691	0.812
Mean (std) of the number of	1.0 (1.1)	1.0 (1.1)	0.9 (0.9)				

comorbidities per encounter					
Hypertension N (%)	128 (42.2)	43 (42.6)	37 (35.6)		
Diabetes N (%)	89 (29.4)	32 (31.7)	30 (28.8)		
Heart Disease N (%)	12 (3.9)	1 (1.0)	2 (1.9)		
COPD N (%)	3 (1.0)	0 (0.0)	1 (1.0)		
Stroke N (%)	1 (0.3)	0 (0.0)	0 (0.0)		
Cerebrovascular Disease N (%)	0 (0.0)	2 (2.0)	0 (0.0)		
Cancer N (%)	4 (1.3)	2 (2.0)	1 (1.0)		
Respiratory Problems N (%)	44 (14.5)	12 (11.9)	15 (14.4)		
Chronic Kidney Disease N (%)	28 (9.2)	11 (10.9)	6 (5.7)		
Tuberculosis N (%)	3 (1.0)	1 (1.0)	3 (2.9)		

Regarding the clinical events used. There were statistically more events in the training cohort (515.958 ± 3,882.337), compared to the testing (186.761 ± 1,217.393) and validation cohorts (176.617 ± 1,133.441), P = 0.014. Conversely, there were no statistically significant differences across encounter types per cohort, P = 0.963; with laboratory events being the most frequent (94%, 94%, and 93% in the training, testing, and validation cohorts, respectively), followed by location (3.6%, 3.3% and 4.3% in the training, testing and validation cohorts, respectively) and vital sign events (0.9 %, 1.2% and 1.2% in the training, testing and validation cohorts, respectively).

4.2.3.2 PROPOSED AND BASELINE MODEL PERFORMANCE

Based on the baseline model development utilizing the UIH Cohort, Random Forest (RF) proved to be the best baseline model. This RF utilized the data features: Age, LDH, O2 Sat, diastolic pressure, AST, creatinine, CRP, ferritin, WBC, RDW, BMI. The summary of the evaluation metrics for both the proposed approach and the baseline models is illustrated in Figure 4-4. Moreover, Table 4-5 shows an evaluation of the sensitivity and specificity for the three models. A t-test of means is performed to test the stated null and alternative hypothesis for both the sensitivity and specificity over the 72-hour time range. This analysis shows that the PM model outperforms both the RF and LR models.

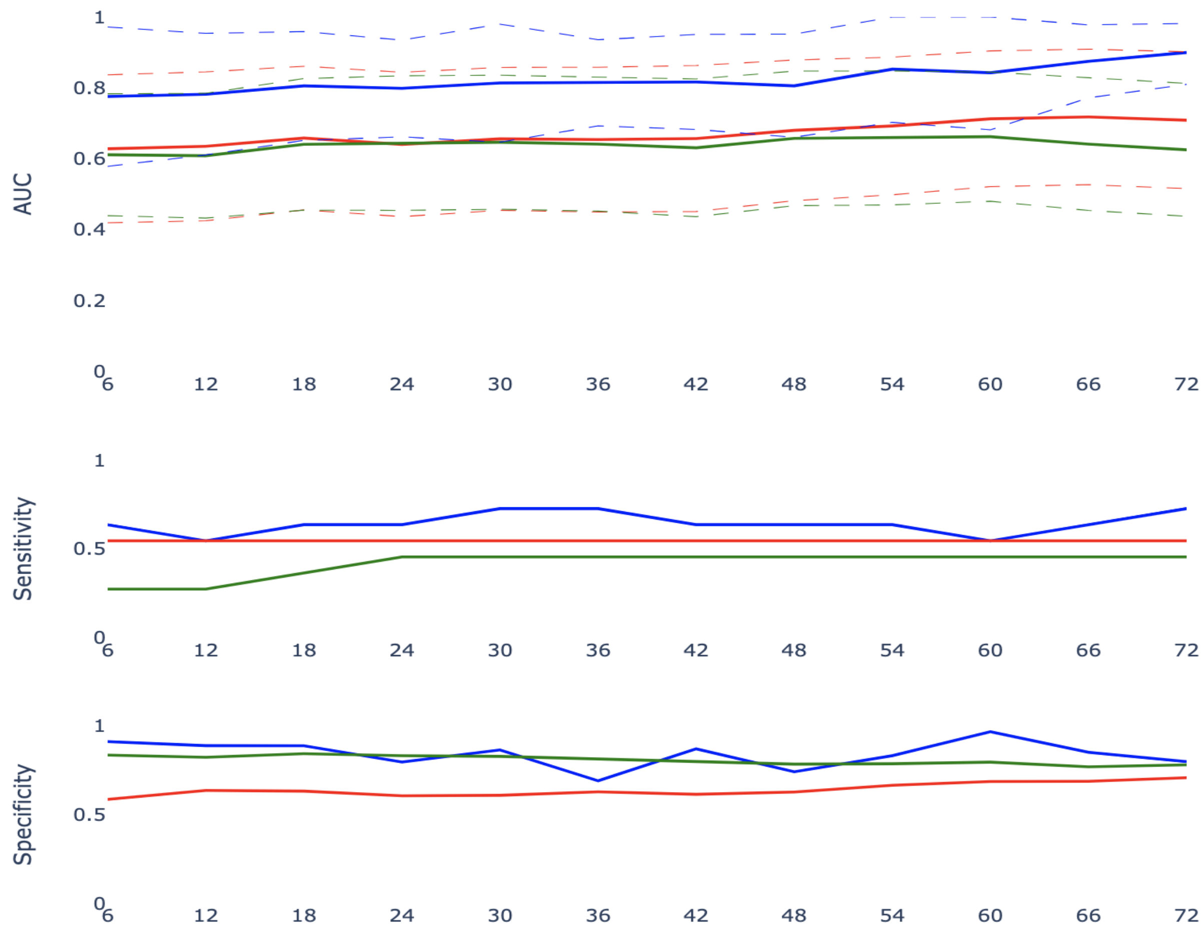

Figure 4-4: Detailed results on the testing cohort

Table 4-5: Statistical comparison of evaluation metrics

Hypothesis		AUC (p-value)
Null	Alternative	
PM=LR	PM>LR	<0.05 (PM has a significantly better AUC than LR)
PM=LR	LR>PM	>0.05 (LR does not have a significantly better AUC than PM)
PM=RF	PM>RF	<0.05 (PM has a significantly better AUC than RF)
PM=RF	RF>PM	>0.05 (RF does not have a significantly better AUC than PM)
RF=LR	RF>LR	>0.05 (RF does not have a significantly better AUC than LR)
RF=LR	LR>RF	>0.05 (LR does not have a significantly better AUC than RF)

4.2.3.3 SHAPLEY VALUE ANALYSIS

Figure 4-5 illustrates the results of the Shapley value analysis for all 6-hour intervals within the first 72 hours of COVD-19 patients. Based on this figure, in almost all cases demographics had the most significant impact on the prediction of mortality in the 6-hour intervals within the first 72 hours after a COVID-19 patient was admitted to the hospital, followed by comorbidities. As we expected age is often a large contributor to the prediction of an outcome of patients with diseases. Older patients tend to be at higher risk of death when having a disease, especially for COVID-19 patients. On the other hand, the impact of other variables was varied from one-time interval to another, and comparing the value of the Shapley analysis for other variables, no consistent order was observed from one-time interval to another. More importantly, the Shapley value analysis also confirmed that the process mining related variables including the time decay function values, markings, and token counts, displayed continuously important roles on the prediction of mortality in the 6-hour intervals within the first 72 hours after a COVID-19 patient was admitted to the hospital and they were presented in all time intervals.

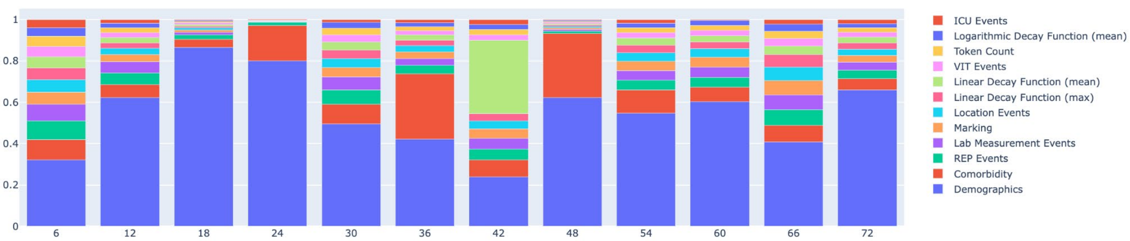

Figure 4-5: Results of Shapley value analysis on each variable for all 6-hour intervals within first 72 hours of admission of COVD-19 patients for the deep learning/process mining approach.

4.2.3.4 DISCUSSION

A significant difference between the process mining, i.e., the DREAM approach, and the baseline models was that progressing time was modeled and leveraged for decision making. Whereas the baseline models used the most recent observations of a patient for outcome prediction, the process mining approach considered the sequence of events over time. Technically, this was realized through the usage of time functions which activated on the observation of events, and which decayed over time [32]. Multiple types of time decay functions were used, such as linear, exponential, and logarithmic time decay functions. Moreover, each of those functions was initialized based on the mean or maximum patient history duration that was observed in the derivation data set.

By following this approach, predictive models can be developed that automatically update the outcome probability based on time. This means that when time progresses, the probability of a patient might change even though no further events have been observed.

The time decay functions values at a given time were fed into a NN, along with event features. Ideally, the NN does not just simply learn the impact of the duration of the last event observation on the outcome probability, but models potentially complex time relationships, such as event interarrival times that influence the outcome probability. In this work, such complex time relationships could be the durations between specific lab measurements, or the duration from admission to ICU in the interplay of performed procedures.

Based on the results of the SHAP analysis, it can be observed that the time decay function values, and the distinct process mining variables such as markings and token counts, demonstrated consistently an important value on the outcome probability. Therefore, it can be concluded that process mining improved our ability to predict the mortality of COVID-19 patients. Hence, contributing time relationships on medical observations of COVID-19 patients were present and should be considered when modeling the data for predictive tasks.

The effectiveness of our developed approach outperformed statistically the best results in the existing literature in terms of the specificity, sensitivity, and AUROC value, and the best baseline model developed by us. Figure 4-4 illustrates the trend changes in AUROC, sensitivity, and specificity among proposed, RF and LR models.

Although the existing proposed methodologies in the literature were successful in predicting mortality of the COVID-19 patients, they possessed several drawbacks. First, RF and LR models were able to predict one class well, and they had poor performance on the other class. Moreover, the machine learning techniques ignored the time information related to the variables which could have a significant impact on the likelihood of mortality.

Our proposed approach had several advantages over prior research papers which are as follows: the process model which was generated through the DREAM algorithm eased the interpretability of a model prediction. An example of a process model can be found in the existing research paper [3], [34]. Moreover, the EHR can be directly used as inputs to our proposed approach without any computationally expensive preprocessing steps. Additionally, the process mining/ deep learning framework could model the time-related variables directly in the prediction algorithm unlike machine learning based models and health calculators.

4.3 PROCESS MINING MODEL TO PREDICT MORTALITY IN ILEUS PATIENTS

This chapter describes a process mining approach to predict mortality in Paralytic Ileus (PI) after 24 hours of the patient being admitted. The chapter is published in a conference proceeding [67].

4.3.1 INTRODUCTION

PI is the obstruction of the intestine due to paralysis of the intestinal muscles [68]. PI prevents the passage of food particles, gas, and liquids through the digestive tract leading to a backlog of food particles impairing digestive movement [69]. MediLexicon International (2018) explains the risk factors include electrolyte imbalance, advancing age, loss of weight, peripheral artery disease, and sepsis. PI is a serious condition and if prolonged and untreated will result in death. Mortality of patients with PI can be as high as 40% in the ICU setting [70]. PI patients who are admitted to the ICU are especially at risk of dying because of the seriousness of their condition [71]. Early prediction of PI could be helpful for clinical decision making and effective usage of medical resources to save patients' lives. Since the results of the existing literature are limited, it is important that current research focuses on developing more accurate models for predicting mortality of ICU patients with PI to increase patients' life expectancy.

Existing literature has developed a variety of predictive modeling to predict the mortality outcome for patients with diseases such as PI [71] and diabetes [47]. Fahad Shabbir Ahmed, et al. [71] developed the SRML-Mortality Predictor framework with two phases. In phase 1, they performed univariate statistical analysis to filter out risk factors or variables which are not significant for predicting mortality for ICU patients with PI after 24 hours of being admitted. The authors conducted cox-regression analysis to provide the hazard ratio about the potential PI risk factors. Moreover, using the Kaplan-Meier analysis they reduced the variables to a reduced-risk factors list of 15 variables, which consists of only the statistically significant risk factors. In phase 2, Ahmed [71] developed multiple machine learning models such as linear discriminant analysis, Gaussian naive Bayes, decision tree model, k-nearest neighbor, and support vector machine (SVM) to predict mortality. SVM led to the best model performance with the AUC score of 81.38%.

Besides machine learning, there is an innovative process mining framework that has shown to be promising for predicting mortality for ICU patients with diabetes [84]. In this work, demographic information of the patients, severity scores on the admission day, and timed state samples which were produced through a process mining approach called DREAM and were fed into a NN to predict mortality which led to an AUC of 87%.

In this current work, we focus on determining the mortality of ICU patients with PI after 24 hours of being admitted, using a process mining modeling approach. The proposed framework is called PMPI (Process Mining Model to predict mortality of PI patients). The PMPI prediction framework demonstrates improved performance in predicting whether a PI patient dies or is discharged after 24 hours of being admitted to the ICU.

4.3.2 APPROACH

This section focuses on the PMPI framework, the proposed method for predicting mortality for ICU patients with PI after 24 hours of admission. First, the feature selection is described. Then, the conversion of EHR to the event logs is explained. In the last part of this section predicting the mortality of ICU patients with PI using the DREAM is introduced. A similar process mining framework was proposed by Theis for predicting in-hospital mortality of ICU patients with diabetes [47].

The PMPI is a modified version of the aforementioned framework and has been customized to handle the mortality prediction of PI patients. Moreover, the PMPI input variables and architecture of the NN are completely different as compared to the said framework.

The PMPI prediction framework is visualized in Figure 4-6.

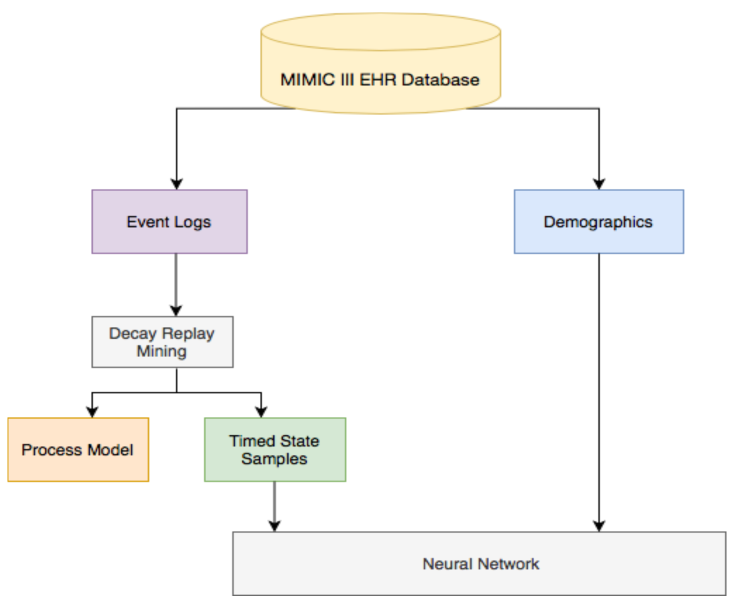

Figure 4-6: Overview of the methodology

4.3.2.1 VARIABLE SELECTION

MIMIC III dataset was used for the PMPI prediction framework. To assure that every patient has a medical history available at the time of prediction, the patients who were admitted at least one time prior to the most current admission were considered. Since the goal is to predict mortality of ICU PI patients after 24 hours of being admitted, the patients who died prior to the 24 hours after being admitted to the ICU were removed.

The prediction is done after 24 hours of the patient being admitted, hence a fair comparison with the existing literature was performed. The PMPI framework is compared with the Fahad Shabbir Ahmed [71] framework for predicting PI patients' mortality. This is the only current work that exists for predicting specifically the mortality of ICU patients with PI. The same data (predictors and target), and the same exclusion and inclusion criteria were used.

4.3.2.2 CONVERSION OF EHR INTO EVENT LOGS

Event logs can be understood as the sequence of events and the associated timestamp at which the events occurred. Each of the events represents performed activities for a patient such as admission, diagnosis, lab measurements, etc., which are known as the careflows of a patient. The patient event logs contain 51 distinct events which were created from the EHR of all patients. Out of 51 distinct events, three of them belong to the admission types, 12 of them are related to the care unit activities and represent the specific location in the ICU patients came in and out after being admitted; CCU, CRU, CSRU, MICU, SICU, and TSICU or left the aforementioned places. Moreover, 34 of the distinct events belong to the lab measurements; 17 are flagged as normal and 17 are flagged as abnormal lab measurements. Finally, two of the events are the type of patient's exit from the system, either death or discharge for each patient. Each unique event has a corresponding timestamp of when the event occurred.

4.3.2.3 PREDICTION

DREAM was used to predict whether a PI patient is discharged from the ICU or dies after 24 hours of being admitted. DREAM replays the event logs on a process model and produces time information which is called timed state samples.

The timed state samples, along with the demographic information of age and insurance types, are fed into the dense NN.

The time state samples are fed into a unique branch of two hidden layers; the first hidden layer has 76 neurons. Following the first layer, a dropout rate is defined for regularization, denoted as DO=0.5. The second hidden layer contains 20 neurons. The demographic information is fed into one single hidden layer of 5 neurons. The aforementioned layers are concatenated into two further layers containing 96 and 10 neurons respectively. A dropout rate of DO=0.5 is defined right after the first hidden layer. An overview of the dense NN architecture for the process mining approach is visualized in Figure 4-7.

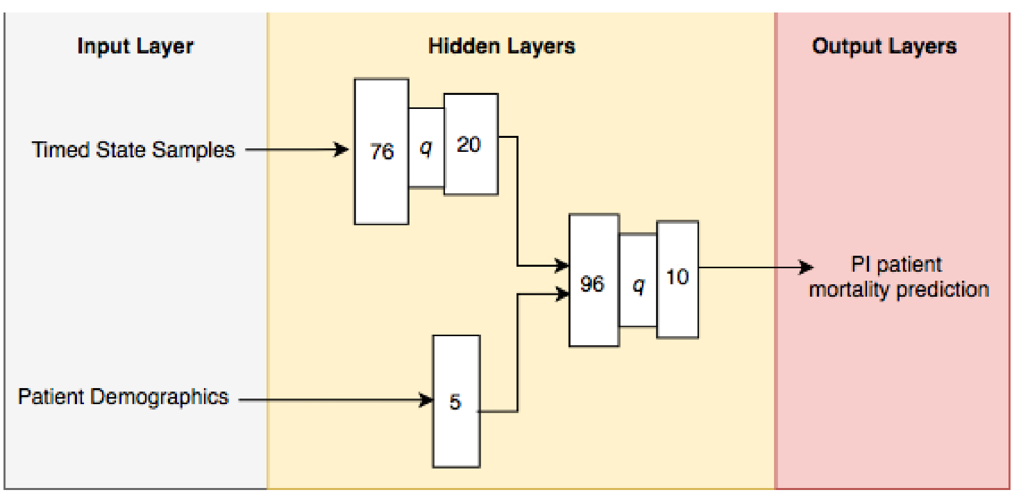

Figure 4-7: An overview of the dense NN architecture for the process mining approach

4.3.3 EVALUATION

This section discusses the experimental evaluation of the PMPI prediction mortality modeling approach for ICU patients with PI after 24 hours of being admitted. In the first subsection, the dataset is described. In the following subsection, the setup for modeling is described for the process model. In the third subsection, the results and comparisons to existing literature are highlighted.

4.3.3.1 DATASET

The data was obtained from MIMIC III. MIMIC III is a large database containing information relating to patients admitted to BIDMC. Data includes vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and more [86]. The data set from ICU admission consisted of 46,476 total patients. A total of 1,067 PI patients were extracted using the ICD-9 code from the MIMIC III database. Furthermore, the PI patients under 18 years of age at their first admission, and who died before 24 hours of being admitted to the ICU were excluded to create a final dataset of 1,017 patients.

Three data types were prepared for the PMPI prediction framework. The first data set was event logs which contained 49 unique events. These events contained EHR information about the patient including admission type, care unit type in and out from the ICU, and normal and abnormal lab items. The second data set consisted of patient demographic data, age, and insurance.

4.3.3.2 SETUP

The dataset of 1,017 patients was randomly split into training and testing sets using a 67/33 ratio producing a train set of 681 patients and a test set of 336 patients. Furthermore, the training set was randomly split using an 80/20 ratio to obtain a train and validation split. The validation sets contained 136 patients. The train and validation sets are required to discover a process model and train the NN. The train and validation set were used to select the best model and the test set was used to evaluate the model performance.

The dataset inclusion and exclusion along with the train, test, and validation split can be visualized in Figure 4-8.

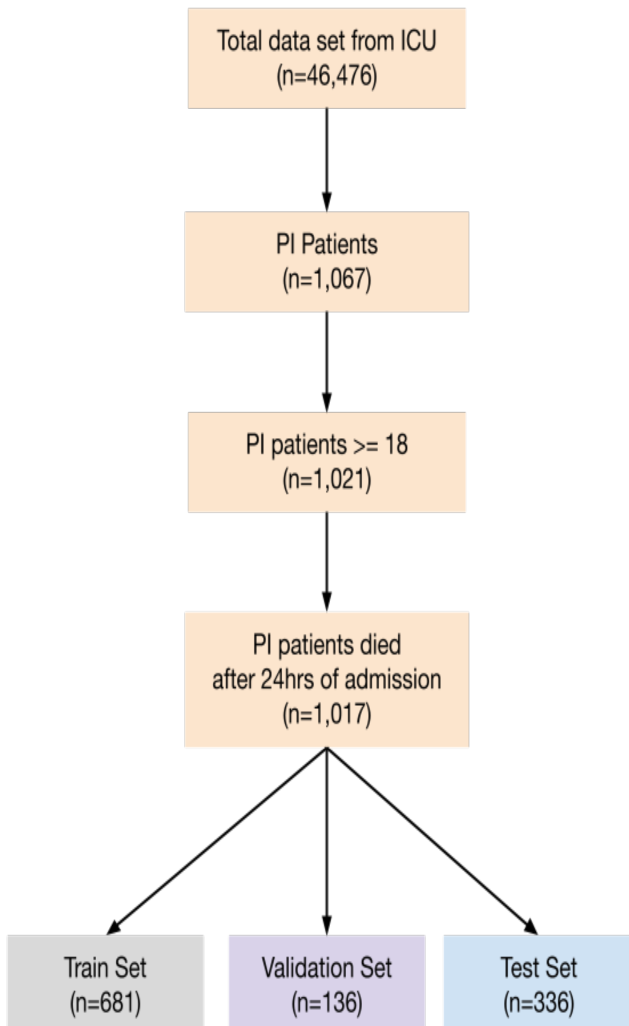

Figure 4-8: Train, test, and validation split

The NN has been trained for 350 epochs using a batch size of 50 with a learning rate of 5e-4 and RMSprop as the optimizer. RMSprop [72] is an optimization algorithm designed for NN.

The metric used was the AUC score. AUC score is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. AUC is a better classification estimate than other common classification performance metrics [88].

The higher the AUC, the better the model is at distinguishing between patients that are discharged and patients that die. Furthermore, the 95% CI for the AUC score was calculated using DeLong's method.

4.3.3.3 RESULTS

The PMPI prediction approach results in the same if not better observed AUC score compared to existing literature which reports an AUC score of 0.81 [71]. The PMPI framework model resulted in an AUC of 0.820 and 95% CI of [0.759, 869] using the test set. In the test set a total of 56 patients actually died, which the model was able to predict 53 of them correctly and misclassified only three of them. On the other hand, 280 patients actually got discharged, which the model was able to predict 120 of them correctly and misclassified 160 patients.

4.3.3.4 SHAPLEY VALUE ANALYSIS

The impact of each feature on the model prediction is evaluated by using SHAP analysis [53]. The SHAP analysis for the features is shown in Figure 4-9 which shows that demographic information predicts mortality for PI patients most after 24 hours of admission. This is because age often contributes to predictions of patient outcomes. Older patients tend to have higher risk of death while having a disease, especially PI patients.

Lab Measurement Types follows as the second most important feature for the prediction of the outcome. Lastly, Admission Types and Care Unit Types are 3rd and 4th important features respectively. The SHAP analysis provides evidence that demographics and patients' medical history information are important contributors to the model prediction.

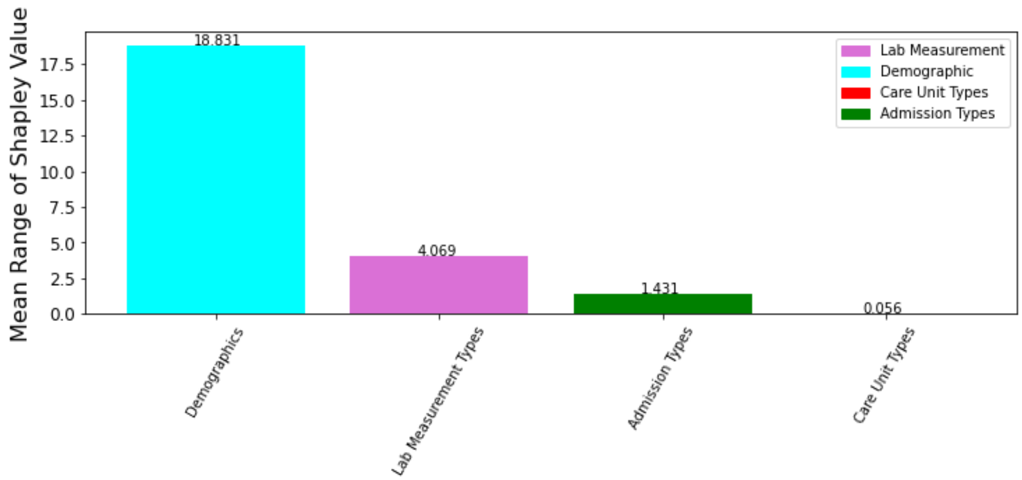

Figure 4-9: The SHAP analysis for the features

One of the main advantages of the process mining approach is that a process model in the form of a PN is discovered. Process mining has been preferred over the state-of-the-art machine learning methods since process mining generates a PN which allows for visualization of the entire processes patients go through in a medical system. While machine learning methods do not allow for visualization, hence interpretation of the results can be difficult. Moreover, process mining has the ability to model the time information related to the variables which other traditional methods are not able to do so.

The PN for the patients in the training dataset contains a total of 22 places and 25 visible transitions which correspond to the events. One of the visible transitions is a merged transition. The black square represents hidden transitions that contain 24 unique events. Hidden transitions also referred to as invisible transitions, represent events that cannot be observed since they have little meaning to representing the process as a whole. A PN allows for visualization of patient careflows for further analysis of PI patients in an ICU setting. Please refer to [67] for PN visualization.

Furthermore, process mining uses the medical history of the patients from prior hospital encounters. Also, the PMPI framework allows for the incorporation of the time information related to the events, timed state samples, as a feature of the NN. Existing models ignore time information; this is a useful attribute for predicting the mortality of PI patients.

A limitation of the proposed approach is that it requires patients' medical history. Unfortunately, small clinics and hospitals might not have records of patients. Also, this method will exclude new patients and patients that have not been admitted to the hospital prior to the current hospital visit. Moreover, hospitals tend not to share patient data across other networks of hospitals. Thus, the proposed approach mostly addresses large hospital networks with patients that have at least 24 hours of hospital medical data.

EFFECT OF PROCESS MINING BASED PRE-PROCESSING STEP IN PREDICTION OF THE CRITICAL HEALTH OUTCOMES

This chapter describes the application of concatenation algorithm which is a pre-processing step explained in Chapter 3, on healthcare datasets to predict critical health outcomes. Also, the process models which were generated through process discovery algorithm were evaluated. The chapter is under review with IEEE Biomedical and Health Informatics.

1.13. INTRODUCTION

A patient's health records can be understood as a sequence of observations including services performed, diagnoses, or lab measurements [32]. In the process mining field, each of these observations is called an event, and the sequences of the observations are referred to as event logs. Process mining analyzes and optimizes processes. The process mining approach has been used widely in the healthcare domains to enhance the healthcare processes [45] or to predict critical health outcomes [32], [47], [67]. The real-life event logs are complex and noisy. Infrequent behaviors and various concurrences generate inefficient and complex process model through process discovery algorithms. Healthcare data exacerbates these issues. For instance, various lab measurements can be taken at the same time causing concurrent relationships among these different lab measurement events. It is therefore critical to pre-process raw event logs to improve its quality, hence making the process model more understandable. A variety of pre-processing algorithms have been used to improve the quality of the data [19], [73]. The concatenation algorithm is a recent pre-processing advance that significantly improves event logs and process model quality for real life benchmark datasets. However, the concatenation algorithm has not yet been tested on real life healthcare event logs. Early prediction of patient health outcomes is critical to clinical decision making and effective allocation of medical resources. For example, CAD causes approximately 610,000 deaths yearly in the United States alone and 17.8 million deaths worldwide [74] Moreover, approximately, 6 million American adults over the age of 20 had HF between 2015 to 2018 [35]. This has resulted in the Centers for Medicare & Medicaid Services penalizing hospitals nearly 2.5 billion USD since 2012 for unplanned 30-day readmission rates [40]. Patient mortality for PI can be as high as 40 in ICU settings [70]. In 2019, diabetes direct caused 1.5 million deaths in 2019. More recently, COVID-19, caused by the SARSCoV-2, heavily impacted morbidity and mortality with major economic consequences [12]. These highlight the need for accurate predictions models to reduce patient mortality rates as well as decrease the readmission by extension financial consequences. Many frameworks have been used to predict the critical health outcomes [33]. DREAM is a recent process mining / deep learning framework. It has significantly improved predicting healthcare outcomes for ICU patients with various diseases [33], [67] using raw, and complex healthcare event logs. However, the effectiveness of pre-processing has not yet been studied using real life healthcare datasets. This work focuses on two results: 1) does pre-processing healthcare datasets improves data quality and enhance the process model; 2) Is the concatenation algorithm an effective pre-processor that predicts critical health outcomes such as patient mortality and readmissions.

1.14. APPROACH

This section focuses on the proposed method for predicting several health outcomes with and without applying a pre-processing step on the event logs. First, feature selection is described. Then, the conversion of EHR to the event logs is explained followed by the preprocessing step. Finally, predicts the health outcomes of the patients using DREAM algorithm is introduced. The intuition behind this methodology is that applying an appropriate pre-processing step on the complex healthcare event logs decreases complexity of the process model. Hence, it predicts critical health outcomes more accurately. Since healthcare event logs contain many concurrences and self-loops, applying concatenation algorithm concatenates some concurrent events to make the process model simpler and generate more accurate predictions. The overview of the proposed framework is visualized in Figure 5-1.

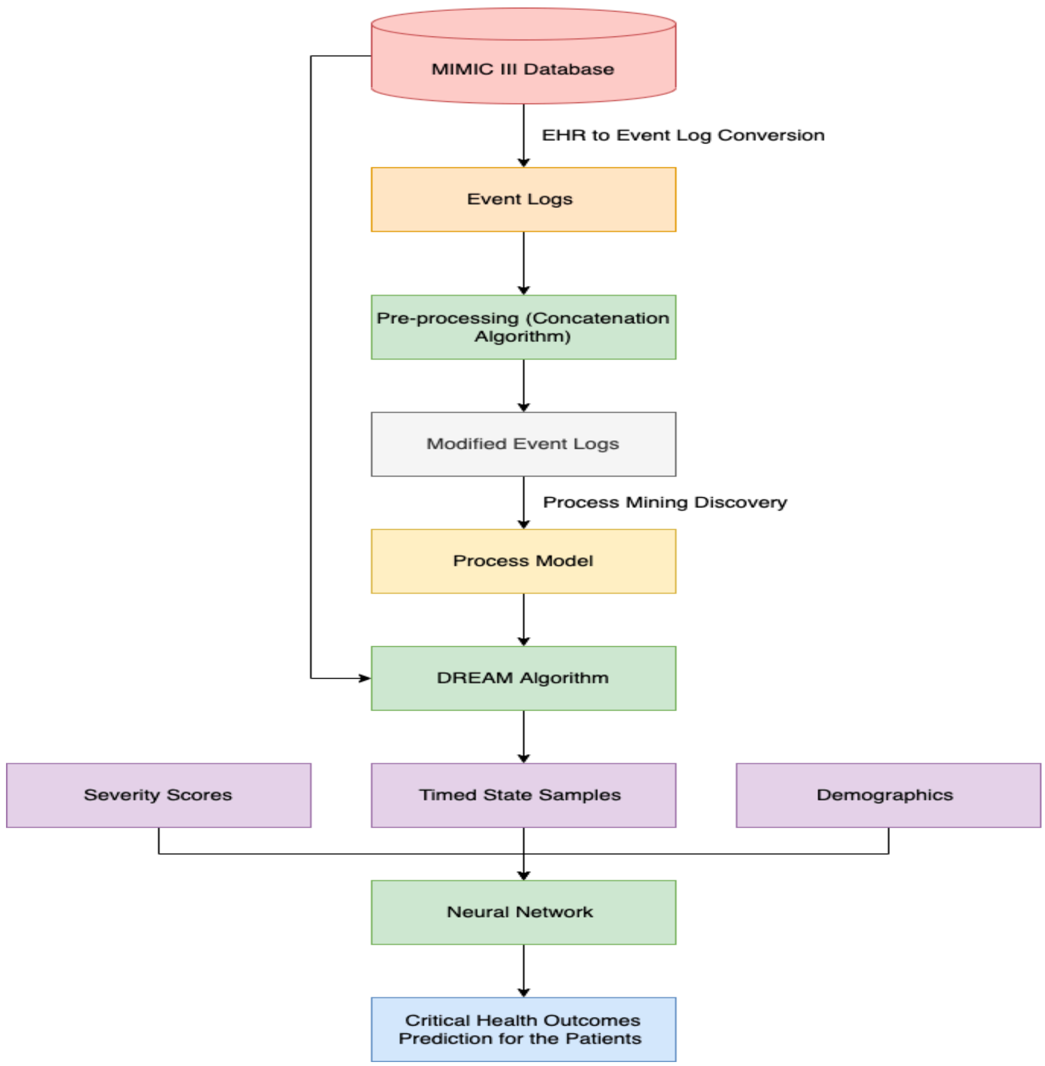

Figure 5-1: Overview of the proposed model

1.14.1. **VARIABLE SELECTION**

MIMIC III and UIH databases were used for predictions. Due to the nature of our approach (process mining- deep learning approach) every patient must have a medical history available at the time of prediction. Table 5-1 shows the features extracted from MIMIC III and UIH databases by disease category:

Table 5-1: Features extracted from MIMIC III and UIH databases

	Data Type									
	Admission	Insurance	Lab Related	Demographics	Elixhauser Comorbidities	CPT and ICD-9-CM codes	Discharge	Severity scores	Care unit in and out from ICU	Location Encounter & report based Vital signs ICU - based
PI	x	x	x	x	NA	NA	x	NA	x	
HF	x	x	x	x	x	x	x	x		
COVID			x	x						X
Diabetes	x	x	x	x	x	x	NA	x	NA	
CAD	r	x	x	x	x	NA	NA	NA	NA	

1.14.2. **CONVERSION OF EHR INTO EVENT LOGS**

Event logs can be understood as the sequence of events with associated timestamps when events occurred. Events represent activities performed on patients such as admission, diagnosis, lab measurements, etc., also known as patients careflows. Each unique event has a corresponding timestamp when it occurred. In this work, the timestamp of comorbidity and artificial events correspond to the discharge time of the corresponding hospital admission. Since events in a trace occur sequentially, multiple comorbidities or artificial events with the same timestamps are delayed by multiples of 1 ms to maintain the order. Table 5-2 summarizes the conversion of the EHR to the event logs.

Table 5-2: Conversion of EHR from databases to event logs

MIMIC III and UIH Tables	Events
ADMISSIONS, PATIENTS	patient's exist type, discharge, insurance type
D_LABITEMS, LABEVENTS, ADMISSIONS	Lab events
1CUSTAYS, CALLOUT	Careunit type
LABEVENTS	Abnormal flagged events
Admissions, Diagnoses_ICD	Elixhauser Comorbidity Score events
Diagnose_ICD, Procedures_ICD, CPTevents	30 Artificial event abstractions

1.14.3. **PRE-PROCESSING STEP**

The resultant event logs (L) from the previous step were then used as an input to the concatenation algorithm. Assume that events e_i, e_j , e_k, and $e_l \in L$, if ($e_i \rightarrow e_j$) and ($e_j \rightarrow e_i$); also ($e_k \rightarrow e_l$) and ($e_l \rightarrow e_k$), there is a concurrent relations between e_i, e_j , $e_i \| e_j$, also between e_k, e_l, $e_k \| e_l$. We then calculate $P(e_i \rightarrow e_j)$ and $P(e_j \rightarrow e_i)$; and $P(e_k \rightarrow e_l)$ and $P(e_l \rightarrow e_k)$. Moreover, threshold value p is introduced, if $P(e_i \rightarrow e_j) > p$ and $P(e_j \rightarrow e_i) > p$, the combination is chosen. Similarly, if $P(e_k \rightarrow e_l) > p$ and $P(e_l \rightarrow e_k) > p$, the combination is chosen for the next step.

Additionally, the probabilities of the selected combinations were added and sorted in the descending order, $P(e_i \parallel e_j) = P(e_i \rightarrow e_j) + P(e_j \rightarrow e_i)$ also $P(e_k \parallel e_l) = P(e_k \rightarrow e_l) + P(e_l \rightarrow e_k)$. Furthermore, if $P(e_i \parallel e_j) = P(e_k \parallel e_l)$ a reposition step was applied. After the order of combinations for concatenation is finalized, the concatenation algorithm selects the ordered combinations one by one for the concatenation. Assume that (e_i, e_j) is chosen for concatenation, event e_i is removed and replaced e_j with the concatenated event, $e_i \oplus e_j$, so the concatenated event will be $e_i \oplus e_j$. This step is repeated until all selected combinations are concatenated in the event logs. After concatenating all combinations in the event logs based on descending order, some events will remain that are not concatenated. Each non concatenated event will be examined and checked if the event, $e_i \in e_i \oplus e_j$. In that case, e_i will be removed and replaced with e_i e_j. If e_i is not part of $e_i \oplus e_j$, while the final trace remains unchanged. Finally, if $e_i \rightarrow e_i$ exist, it is removed. The modified event logs were then fed to the SM algorithm to produce a process model. The resultant process model used as an input to the DREAM algorithm for prediction which the details are explained in the next subsection.

1.14.4. **PREDICTION**

DREAM was used to predict the critical health outcomes for both pre-processed and non-processed event logs. DREAM replays the event logs on a process model and produces time information related to the variables which are called TSS. The TSS, demographic information, and severity scores were fed into the dense NN for prediction. The NN architecture differs by disease but remained the same for both pre-processed and unprocessed event logs.

1) CAD disease: TSS are fed into a unique branch of one hidden layer with 200 neurons. The DO rate after the first layer was regularized at DO = 0.2. Demographic information was fed into a single hidden layer with 20 neurons and a dropout rate of 0.2. These were concatenated into a second layer with 90 neurons and a dropout rate of DO = 0.2. The critical outcome which is predicted in this case is mortality of ICU patients with CAD.

2) HF disease: TSS, demographics information, and severity scores were fed separately to three branches, each with three hidden layers. A batch normalization layer was added after the first hidden layer for each branch with a dropout rate of 20%. The output layer included a softmax activation function to predict unplanned 30-day readmission of ICU HF patients. The critical outcome which is predicted in this case is unplanned 30-day readmission of ICU patients with HF.

3) Diabetes: each input layer is fed to an individual hidden layer branch before a hidden layer concatenates the branches and feeds it to two further layers. All hidden layers use a ReLU. The branched first hidden layers have additionally a batch normalization layer. Dropout layers with a rate of 40% for regularization are used. The output layer consists of a softmax activation function to output the patient's probability of in-hospital mortality for diabetes patients.

4) COVID- 19: TSS, demographics information and comorbidities were fed separately to two branches which first branch contains three hidden layers with 90, 50 and 20 neurons respectively. The first and second hidden layers each had a dropout layer with a rate of 20%. The second branch also contained one hidden layer with 5 neurons. The two branches were then concatenated to a branch with three hidden layers, containing 90, 50, and 20 neurons respectively. The dropout layer after the second concatenated hidden layer had a rate of 30%. The output layer used a softmax activation function to predict mortality of COVID- 19 patients. The outcome which is predicted is mortality every 6-hour during the first 72 hours after hospital admission.

5) PI: TSS were fed into a unique branch of two hidden layers. The first had 76 neurons and a subsequent dropout rate of DO = 0.5. The second hidden layer contains 20 neurons. The demographic information is fed into a single hidden layer of 5 neurons. Both layers were concatenated into two further layers with 96 and 10 neurons respectively. Between these two layers, there is a layer with a dropout rate of DO = 0.5. The outcome which is predicted in this case is the mortality of ICU patients diagnosed with PI after 24 hours of being admitted to the ICU.

1.15. EVALUATION

This section discusses the experimental evaluation of the model to predict the critical health outcomes. Subsection one describes the datasets. Next, subsection two describes the modeling set up. Finally, subsection three highlights the results.

1.15.1. DATASET

The data was obtained from MIMIC III [75], a large database containing information relating to patients admitted to BIDM. This data includes vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and more [20]. Also, the data was obtained from UIH which is a tertiary, teaching hospital in Chicago. This study was approved by The UIC Institutional Review Board. A summary of the datasets for each disease category is shown in Table 5-3.

Table 5-3: Summary of datasets by disease category'

Disease	Total Numbers of Patients	Train	Test	Validation
HF	3,411	2,422	555	434
COVID-19	508	303	104	101
PI	1,153	681	336	136
CAD	2,176	1,202	674	300
Diabetes	2,435	1,552	609	274

1.15.2. SETUP

RapidProM was used to run the experiments [30] which extends RapidMiner with process mining analysis capabilities. The concatenation algorithm's threshold value of p was used to select the combinations of concurrent events for further concatenation steps. The optimal value of p was based on the highest value of the F-Measure. The hyper-parameter optimization was conducted on 16 datasets using steps of 0.1. Of these, 13, led to finding the optimal value of p = 0:7. SM takes two threshold values: Ƞ and ε. These values were optimized based on the highest F-Measure values using steps of 0.1. Thresholds selected were Ƞ = 0. 4 and ε = 0. 1. Train and validation sets are required to discover a process model and train the NN. The validation set is used to select the best model and test set is used to evaluate model performance. The summary of the set up for training the NN in each disease category is shown in Table 5-4. We measured the quality of the process models with and without pre-processing using common metrics of fitness, precision, and F-Measure as proxies for accuracy, size, CFC, and structuredness, which indicate complexity. The metric used to evaluate the model prediction is the AUC score which is equal to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one. AUC is a better classification estimate than other common classification performance metrics [76]. Higher AUC scores indicate that a model is better at distinguishing between discharged patients and those who die. Furthermore, the 95% CI for the AUC score is calculated using DeLong's method [53].

Table 5-4: Summary of setup for training NN

Disease	Activation Function	Epoch	Batch Size	Optimizer
HF	ReLU	100	12	Adam
COVID-19		350	10	Adam
PI		350	50	RMSprop
CAD		300	56	RMSprop
Diabetes		200	256	Adam

1.15.3. RESULTS

In our first experiments without pre-processing, we used the optimal threshold values found for the SM algorithm, plus raw event logs to discover a process model. The model was fed to the DREAM algorithm to produce the TSS. The TSS, severity score and demographics were then fed to a NN for prediction. For our second experiments, with pre-processing we applied the concatenation algorithm to raw event logs using optimal thresholds values for the concatenation algorithm to produce the modified event logs. The modified event logs were then fed to the SM algorithm with the aforementioned optimal threshold values to discover a process model, which was then fed to the DREAM algorithm to produce the TSS. The TSS, severity score and demographics were then fed to a NN for prediction. Results of the evaluation are summarized in Tables 5-5, 5-6, and 5-7. We ran the Wilcoxon Test to determine whether improvements observed for evaluation metrics applied after concatenation were statistically significant or not ($p = 0.05$). Of the 16 datasets were pre-processed, 9 datasets showed statistically significant differences in AUC scores. Moreover, all 16 showed statistically significant differences in F-Measures and complexity metrics.

Table 5-5: Results in terms of F-Measure on event logs, and the Wilcoxon test results on improved cases to statistically identify significance level.

	Evaluation Metrics F-Measure		
Type of Disease	**before concat**	**after concat**	**p -value**
HF	0.821	0.830	0.048
PI	0.842	0.861	0.039
COVID 6hr	0.844	0.871	0.037
COVID 12hr	0.844	0.871	0.037
COVID 18hr	0.844	0.871	0.037
COV ID 24hr	0.844	0.871	0.037
COVID 30hr	0.844	0.871	0.037
COV1D 36hr	0.844	0.871	0.037
COVID 42hr	0.844	0.871	0.037
COVID 48hr	0.844	0.871	0.037
COVID 54hr	0.844	0.871	0.037
COVID 60hr	0.844	0.871	0.037
COVID 66hr	0.844	0.871	0.037
COVID 72 hr	0.844	0.871	0.037
Diabetes	0.791	0.792	0.055
CAD	0.832	0.854	0.045

Table 5-6: Results of the proposed model in terms of AUC and CI on event logs, and the Wilcoxon test results on the improved cases to statistically identify the level of significance

	Evaluation Metrics				
	AUC			**CI**	
Type of Disease	**before concat**	**after concat**	**p-value**	**before concat**	**after concat**
HF	0.930	0.947	0.045	[0.898- 0.960]	[0.910- 0.971]
PI	0.810	0.823	0.047	[0.768- 0.840]	[0.788-0.850]
COVID 6hr	0.776	0.779	0.055	[0.678- 0.876]	[0.701-0.790]
COVID 12hr	0.782	0.79	0.043	[0.685- 0.880]	[0.708-0.850]
COVID 18hr	0.806	0.802	NA	[0.719-0.901]	[0.788-0.850]
COVID 24hr	0.799	0.791	NA	[0.698- 0.890]	[0.718-0.870]
COVID 30hr	0.814	0.841	0.034	[0,718- 0.910]	[0.786-0.855]
COVID 36hr	0.814	0.799	NA	[0.718- 0.900]	[0.778-0.890]
COVID 42hr	0.817	0.817	NA	[0.701-0.870]	[0.789-0.860]
COVID 48hr	0.806	0.82	0.042	[0.738- 0.890]	[0.784-0.880]
COVID 54hr	0.853	0.869	0.039	[0.758- 0.910]	[0.788-0.890]
COVID 60hr	0.843	0.842	NA	[0.768- 0.8701	[0.748- 0.890]
COVID 66hr	0.875	0.881	0.046	[0.778- 0.9401	[0.798-0.950]
COVID 72hr	0.900	0.915	0.045	[0.868-1.00]	[0.870- 0.990]
Diabetes	0.873	0.861	NA	[0.851- 0.940]	[0.788-0.890]
CAD	0.871	0.890	0.044	[0.831- 0.913]	[0.870-0.923]

Table 5-7: Results of in terms of complexity metrics on event logs, and the Wilcoxon test results on the improved cases to statistically identify the level of significance

	Evaluation Metrics								
	Size			Struct.			CFC		
Type of Disease	**before concat**	**after concat**	**p-value**	**before concat**	**after concat**	**p-value**	**after concat**	**before concat**	**p-value**
HF	70	46	0.032	142	121	0.034	71	66	0.032
PI	151	89	0.037	301	190	0.036	88	67	0.035
COVID 6hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 12hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 18hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 24hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 30hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 36hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 42hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 48hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 54hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 60hr	1001	401	0.031	2315	998	0.033	142	71	0.031
COVID 66hr	1001	401	0.031	2325	998	0.033	142	71	0.031
COVID 72hr	1001	401	0.031	2325	998	0.033	142	71	0.031
Diabetes	68	35	0.034	121	101	0.043	70	69	0.035
CAD	98	79	0.032	132	97	0.031	73	66	0.038

1.16. DISCUSSION

To reduce concurrences and self-loops of the complex healthcare data, we applied a concatenation algorithm as a pre-processing step to improve the quality of the data, hence improve trustworthiness of the prediction modeling for clinicians. We observed significant statistical improvements in AUC values, F-Measure and complexity metrics. This was especially evident for the CAD and COVID-19 datasets which were more complex compared to other datasets. In their cases when we ran the experiments, the predicted results improved by significant statistical amount of 2% for the AUC metric. One of the main advantages of the process mining approach is that it discovers a process model which allows clinicians to visualize the entire processes patients (careflow) go through in a medical system, which is why process mining has been preferred over state-of-the-art methods. Process mining also models time information related to variables and uses them as an input to the NN that other traditional methods cannot. Lastly, process mining uses the medical history of patients from prior hospital admissions. A limitation of the proposed approach is that it requires patients' medical history which small clinics and hospitals might not have. It also excludes patients with no prior hospital admissions like new patients and patients not admitted before the current hospital visit. Moreover, hospitals tend not to share patient data across outside hospital networks. Thus, the proposed approach best matches large hospital networks. In addition, the train, validation, and test sets all came from the MIMIC-III and UIH datasets. Therefore, using an independent dataset from a different system would better test the performance of the model for future research.

REDECA: A Novel Framework to Review Artificial Intelligence and Its Applications in Occupational Safety and Health

This chapter describes a framework to review the application of AI in OSH in main industry sectors. The chapter is obtained with permission of Environmental Research and Public Health journal from the previously published work, "REDECA: A Novel Framework to Review Artificial Intelligence and Its Applications in Occupational Safety and Health" [77].

1.17. INTRODUCTION

AI is an extensive and diverse research field that has infiltrated every aspect of our lives and gained decisive importance over the years with over 20,000 publications in 2019 alone (Figure 6-1) [78] In basic terms, AI is the ability of a computer to process information and generate outcomes that mimic how a human learns, makes decisions, and solves problems [79]. While research in AI is relatively new, the concept of AI can be traced back to as early as the 1940s where Alan Turing was one of the first mathematicians to explore the mathematical possibility of AI by posing "whether a machine can think like a human or not" [80]. The term "artificial intelligence" was proposed in a series of workshops at the Dartmouth Summer Research Project on Artificial Intelligence (DSRPAI) hosted by John McCarthy and Marvin Minsky in 1956 [81]. Academia and industry have applied AI to solve various problems such as decision making [82], environmental monitoring [83], [84], lower operational costs [85], and increase productivity [86]. The advent of technological advances in robotics, sensors, data management, and computer technology on one hand, and powerful machine learning algorithms, on the other hand, have opened vast opportunities to apply AI in various fields (Figure 6-2). For example, machine learning algorithms are being used to: optimize the performance of a network of sensors used for detecting moving objects [87], select the location of radio frequency sensors used by police/firemen to detect indoor crews in the event of a fire or other threats [88], detect vocal disorder among workers who use their voice maneuvers extensively such as singers and teachers [89], and used to predict bankruptcy [90]. Other important AI applications include facial recognition technology for law enforcement [91], improvement in marketing and customer service [92], and dramatic improvements in the accuracy of digital imaging [93], [94]. These studies point to accumulating evidence that AI technology could effectively be used to detect, identify, and predict risky behavior in a potentially hazardous working environment.

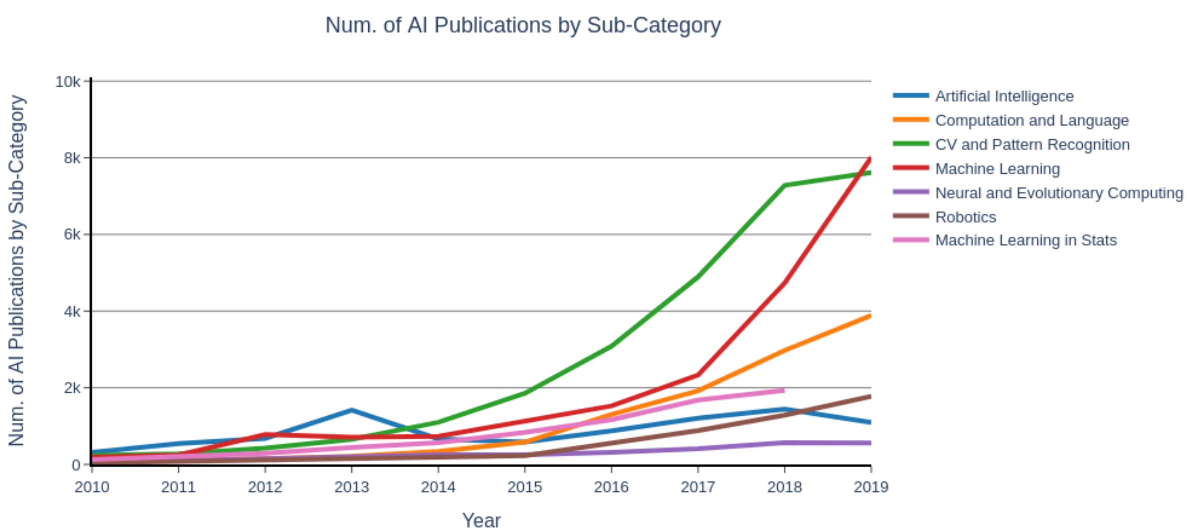

Figure 6-1: Number of AI papers on ArXiv by subcategory (y-axis) from 2010 to 2019 (x-axis) [78].

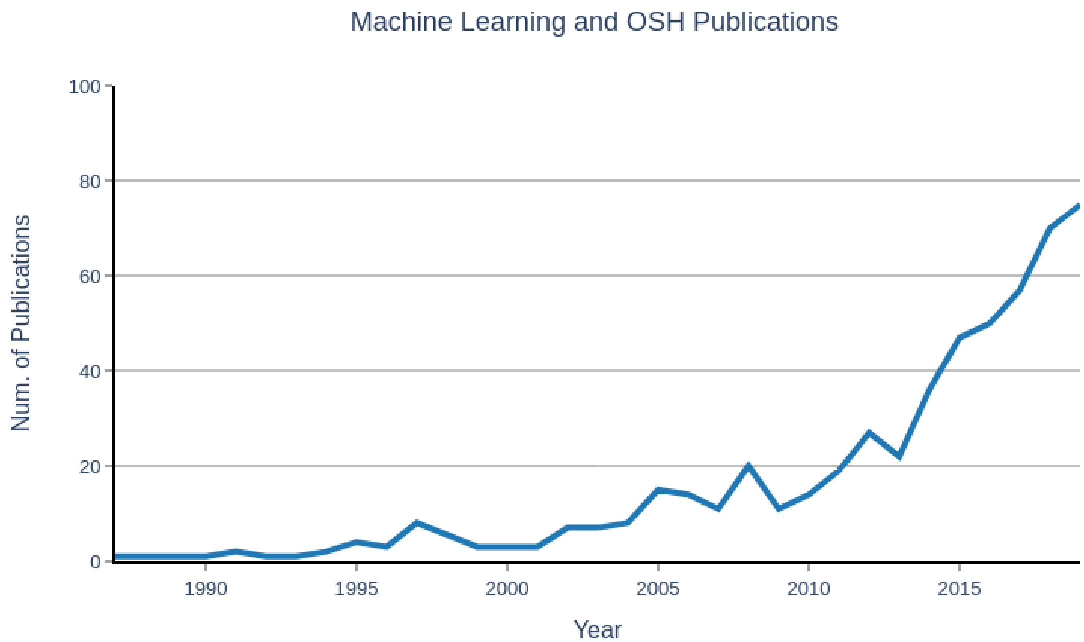

Figure 6-2: Number of publications (y-axis) applying AI to the OSH field from 1986 to 2019 (x-axis). All AI papers queried were individually reviewed to confirm OSH application.

1.18. APPROACH

1.18.1. Risk Evolution, Detection, Evaluation, and Control of Accidents (REDECA) Framework

Figure 6-3 describes a novel framework called REDECA developed by the authors to theorize how AI technologies and methods can be used to anticipate and control the risk of exposure in a worker's immediate environment. The REDECA framework is constructed based on the fundamental underlying idea of the Swiss cheese model [95] that is a dominant paradigm for depicting how injury incidents in complex systems occur. Based on this model a given hazard can generate a safety incident when multiple layers of defenses and safeguards (or interventions) designed to prevent the incident or loss fail to properly act. While the Swiss cheese model conceptualizes that a safety incident occurs when multiple stages of safeguards fail, it is not capable of showing how AI can be used in each step of this process. To create this capability, we extend the Swiss cheese model by including new details that are necessary to describe how AI can be and has been used in detecting, preventing, and controlling the evolution of safety accidents. These details include the characteristics of each state visited when reaching from a safe state to an accident state, the probabilities and timing information associated to each state, and the interventions that can reverse or slow down such a process. We have shown all these details by the REDECA framework shown in Figure 6-3. We assume that a human worker, due to the nature of his work, can be at different levels of safety risk at any given time. These levels are shown by the three states of R1, R2, and R3 (shown by blue boxes in Figure 6-3). R1 is the ideal state where a worker has minimal to no risk of exposure to the hazard. Our goal as OSH professionals is to keep the worker in this state. However, this is often not achievable due to the work requirements, available technologies, environmental factors, budget, etc. In R2 the worker is at an increased risk of a harmful work-related exposure event but has not experienced a harmful event. R3 is the state when a harmful work-related event has already occurred impacting the health and safety of the worker. AI technology-based inputs can monitor and foresee the change in the state of risk and impact movement between these states of risk to minimize damage from a harmful work-related event. To minimize the chance and negative consequences of safety incidents, we are interested in three types of information and actions related to the states R1, R2, and R3: 1—transition probabilities and times for moving from a lower risk state to a higher risk state (green boxes in Figure 6-3), 2—detection of a state change (white boxes in Figure 6-3), and 3—interventions in each state that reduce the risk level or negative consequences of safety incidents (orange boxes in Figure 6-3).

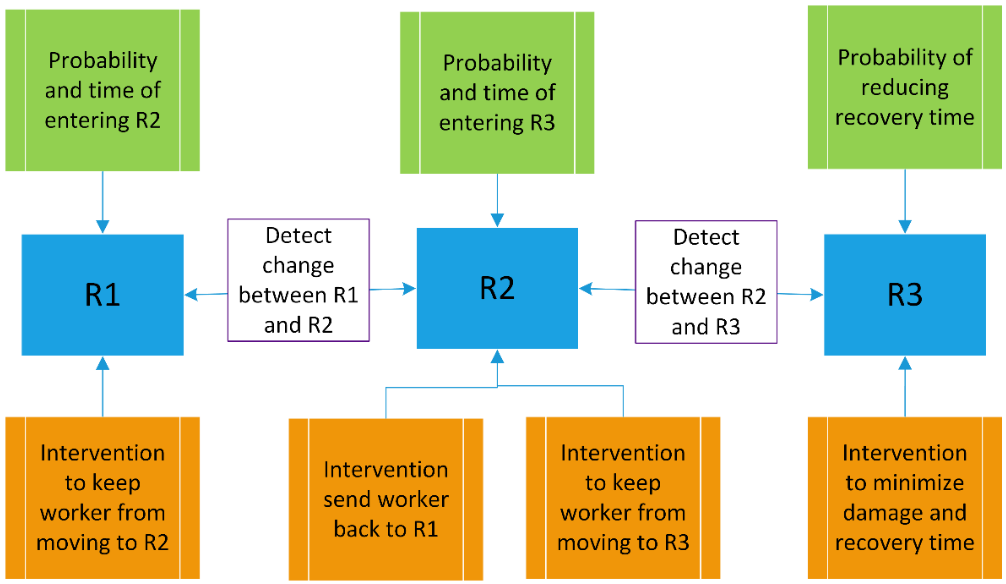

Figure 6-3: The REDECA framework for AI OSH

Blue boxes are different states where workers can find themselves. In state R1 worker risk is minimal or no risk of exposure. R2 indicates exposure to hazard and increased risk of injury. R3 indicates a harmful work-related event occurred. Green boxes indicate technologies that can predict the probability of transitioning into next states. White boxes are technologies that can detect transitions between states. Orange boxes indicate intervention strategies to keep worker safe or reduce the impact of a work-related event.

For a worker who is in state R1, we are interested in AI based models and technologies that help us with the followings: calculating the probability and/or time left for the worker to transition from the safe state (R1) to the hazard-exposed state of R2; detecting (sensing) the event that shows such a transition; and designing and implementing AI based technologies that keep the worker in R1 or at least reduce the probability of moving from R1 to R2.

For a worker in state R2, we are interested in the AI models/technologies that assist us with the followings: calculating the probability and time left for a worker transition from the hazard-exposed state of R2 to R1 or R3; detecting or sensing the events corresponding to these transitions;

and the design and implementation of AI technologies and models that could send the worker back to the R1 state or at least reduce the probability of having a safety incident, i.e., moving from R2 to R3.

If a worker experiences an injury incident, then the worker's state is set to R3. In this state, we are interested in AI models/technologies that help in reducing the damage and recovery time of the worker, and in calculating the times and probability of recovery.

All the AI/OSH papers reviewed by the authors are related to at least one of the green, white, or orange boxes shown in Figure 6-3. Therefore, we use this framework to classify AI/OSH literature related to worker's safety in the five industries of agriculture, oil and gas, mining, transportation, and construction.

1.18.2. **Literature Search Strategy**

The five most dangerous industries by fatal injuries are agricultural, mining, oil and gas, transportation, and construction respectively [96]. In 2019, according to the U.S. Bureau of Labor Statistics these industries experienced almost 2700 fatal injuries, 50% of all fatal injuries reported that year. These industries also had over 204,000 injuries that resulted in days away from work, approximately 25% of all injuries in 2019. Moreover, these industries had the highest fatal injury rates of all other industrial sectors and were chosen for this review work (Figure 6-4); [96]. The application of AI and machine learning algorithms, actuators and sensors in the OSH field for these industries were reviewed by using PubMed, Google Scholar, and Scopus search engines to find relevant research. Different keywords such as "artificial intelligence", "occupational safety and health", "agriculture", "mining", "oil and gas", "construction", "transportation", "ergonomic", "risk factors", "sensors devices", "robots" and their combinations were used to explore available papers in the fields of AI and OSH. For each selected paper, a backward and forward citation search was conducted to capture additional papers not found in the original queries. Over 650 abstracts were reviewed and only papers that were non-repetitive, English-based, relevant to OSH, AI and the five industrial sectors were chosen for further review. The full text of the remaining publications was then read and only papers that meet the criteria specified in our REDECA framework and within the five industries (agriculture, mining, oil and gas, transportation, and construction) were included in this work.

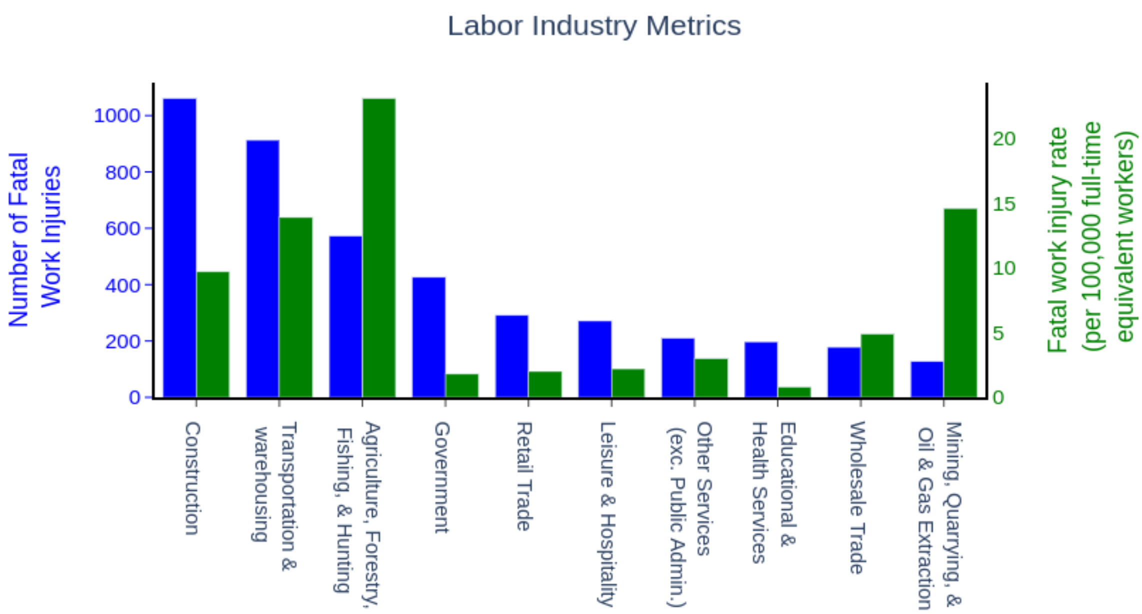

Figure 6-4: U.S. Bureau of Labor Statistics 2019 survey on the rate of fatal work injury by industry sector [96].
y-axis indicates the rate of fatal work injury, while x-axis shows different industries.

Each paper was reviewed and classified using the REDECA framework and AI system. The algorithms, sensors, actuators, and environment used/described by the paper were organized by industry in tables highlighting where the majority of AI research in each industry is located within the REDECA framework and AI system. Each component of the REDECA framework was used in the tables using shorthand descriptions and components where there was no available research included in the tables to highlight potential research gaps (Table 6-1).

Table 6-1: REDECA components and shorthand notations used in industry

Prob. R2	Probability and time of entering R2
Detect R1→R2	Detect change between R1 and R2
Int. R1→R2	Intervention to keep workers from moving to R2
Int. R2→R1	Intervention send worker back to R1
Prob. R3	Probability and time of entering R3
Detect R2→R3	Detect change between R2 and R3
Int. R2→R3	Intervention to keep workers from moving to R3
Prob. Rec.	Probability of reducing recovery time
Int. R3	Intervention to minimize damage and recovery time

1.19. RESULTS

Tables 6-2 – 6-6 summarize the number of papers published relating AI interventions, by category, to the AI approach in the REDECA framework. These highlights the strengths, opportunities, and weaknesses of using AI in OSH.

Table 6-2: Agricultural AI/OSH algorithm, sensor, and actuator research organized by REDECA framework.
Values indicate the number of papers using AI algorithms, sensors, or actuators.

REDECA Components	AI Algorithms Machine Learning (ML)	Sensors	Actuators
Prob. R2	4	6	7
Detect R1→R2	0	4	1
Int. R1→R2	2	14	10
Int. R2→R1	0	0	0
Prob. R3	0	4	0
Detect R2→R3	0	0	0
Int. R2→R3	0	0	0
Prob. Rec.	0	0	0
Int. R3	0	0	0

Table 6-3: Oil and Gas AI/OSH algorithm, sensor, and actuator research organized by REDECA framework.
Values indicate the number of papers using either AI algorithms, sensors, or actuators.

REDECA Component	AI Algorithms (ML)	Sensors	Actuators
Prob. R2	7	11	0
Detect R1→R2	6	20	1
Int. R1→R2	1	9	0
Int. R2→R1	0	0	0
Prob. R3	0	0	0
Detect R2→R3	0	0	0
Int. R2→R3	0	0	0
Prob. Rec.	0	0	0
Int. R3	0	0	0

Table 6-4: Mining AI/OSH algorithm, sensor, and actuator research organized by REDECA framework.
Values indicate the number of papers using either AI algorithms, sensors, or actuators.

REDECA Component	AI Algorithms (ML)	Sensors	Actuators
Prob. R2	0	0	2
Detect R1→R2	0	29	6
Int. R1→R2	0	5	2
Int. R2→R1	0	11	2
Prob. R3	0	12	3
Detect R2→R3	0	0	0
Int. R2→R3	0	0	0
Prob. Rec.	0	0	0
Int. R3	0	0	0

Table 6-5: Transportation AI/ OSH algorithm, sensor, and actuator research organized by REDECA framework.
Values indicate the number of papers using either AI algorithms, sensors, or actuators.

REDECA Component	AI Algorithms (ML)	Sensors	Actuators
Prob. R2	3	6	0
Detect R1→R2	20	23	3
Int. R1→R2	3	7	3
Int. R2→R1	0	0	0
Prob. R3	2	4	0
Detect R2→R3	0	0	0
Int. R2→R3	0	0	0
Prob. Rec.	0	0	0
Int. R3	0	0	0

Table 6-6: Construction AI/ OSH algorithm, sensor, and actuator research organized by REDECA framework.
Values indicate the number of papers using either AI algorithms, sensors, or actuators.

REDECA Component	AI Algorithms (ML)	Sensors	Actuators
Prob. R2	1	2	0
Detect R1→R2	2	2	0
Int. R1→R2	0	0	6
Int. R2→R1	1	0	0
Prob. R3	1	1	0
Detect R2→R3	30	37	0
Int. R2→R3	0	0	0
Prob. Rec.	0	0	0
Int. R3	1	1	1

1.20. DISCUSSION

The application of AI in the realm of several industries has been described as the Fourth Industrial Revolution [97]. Innovations in artificial intelligence using sensors, robots, machine learning algorithms have been shown to increase productivity and could potentially improve the safety and health of workers in the workplace. Since the application of AI in workplaces has increased over the past few years, it is very crucial to have a thorough understanding of AI methods, and the effects of these methods on the workers and workplaces as well. To aid in this understanding, this work developed the REDECA framework to categorize and highlight the applications of AI in OSH. This novel approach is a natural by-product of the literature developed. It was created by carefully reviewing the literature and developing large categories where the papers in the literature fell. The available OSH AI literature was compiled in tables by industry and by AI system element to identify the key strengths, weaknesses, and opportunities.

In brief, the construction industry and evaluating driver fatigue in the transportation industry had many AI algorithms identified in the peer-reviewed literature. These algorithms spanned across most elements of the AI REDECA. Conversely, in agriculture, mining, and oil and gas industries there were very few AI algorithms used. Similarly, we see the agriculture and mining industries have many actuators when other industries did not. In all industries, there were many papers published describing the use of sensors and environment descriptors. The ability to be able to quickly view where there are gaps in the literature across the AI system is the strength of using this framework. Another strength of the program is to be able to identify which part of the REDECA is missing AI involvement.

By separating the papers in the published literature into their targeted approach to protecting workers using the AI REDECA it becomes clear that most AI interventions target probabilities, detection, and interventions when a worker is in R1. In general, there is a lack of developed and published material describing AI systems aimed at detecting when someone goes from being exposed to a risk environment (R2) to being injured or put in risk state 3. This then precludes one from establishing how long it will take to return to health. There is also an opportunity to develop AI models targeting interventions to keep workers from moving to R3 and interventions to minimize the damage from being in R3 and improve recovery time. When protecting workers, it is important to focus efforts on the early stages of intervention with the goal of never having a worker in R3. Unfortunately, this is not always possible and thus the opportunity uncovered by using this framework is to develop AI systems targeted at reducing the probability and increasing the interventions of workers in R3 of the AI REDECA. These elements are crucial to minimizing harm in the event of workplace incidents.

This work is not a systematic review of the AI literature. Our work is the first survey of the reach of existing applications of AI in OSH and documents several examples of how AI can enhance the effectiveness of OSH interventions to protect workers in diverse work sectors. The authors acknowledge the limitations of the current work and recommend several areas for further exploration:

1. First and foremost, a systematic review of scientific journals, industry reports, and other practice journals may provide insights into more applications of AI in OSH beyond the scope of this survey. Additionally, qualitative approaches may be needed to fully understand the dynamics of AI-OSH teams in the field that have not been captured in this survey.

2. Our survey did not find any educational papers about the AI curriculum or training in OSH. A recent work specifically highlights the need for OSH professionals, practitioners, researchers, employers, and workers should develop a better understanding of worker health, safety, and well-being applications of AI [98]. A comprehensive scan of existing AI curricula in academia and training and skills needs among OSH professionals in the industry may provide a better understanding of future AI capacity needs for OSH researchers and practitioners. For example, a significant increase in the availability of funding for AI applications in healthcare over the past ten years has led to a shift in the number of students and healthcare professionals with access to AI training and the capacity to implement AI applications.

3. Currently, there is no dedicated funding source for AI research or practice in OSH. The fourth industrial revolution (also known and Industry 4.0) is here and the NIOSH Future of Work Initiative was launched in 2019 to identify novel research solutions, practical approaches, and stakeholder opportunities to collectively address the future of work [99], [100]. AI, including deep learning, neural networks, and machine learning, are priority topics and subtopics listed in the guiding framework for NIOSH research and practice-based activities as part of this initiative [99]. We need to advocate for resources to fund the research and training of OSH professionals in

governmental agencies (NIOSH), academic institutions, and industries to fully leverage the capacity of AI to protect the health, safety, and well-being of workers.

4. AI will continue to play a very significant role in the design of future workplaces, work health, and worker well-being. It is anticipated that massive innovation in industries driven by AI could potentially lead to the creation of new sectors for growth and jobs and eliminate several existing jobs. Recently the European Commission proposed new rules and actions aiming to “turn Europe into the global hub for trustworthy Artificial Intelligence (AI)” [100]. The goal is to “coordinate a plan with the Member States to ensure the safety and fundamental rights of people and businesses while strengthening AI uptake, investment, and innovation across the EU” [100]. This aspect of AI was not the focus of this work, but the authors recognize the potential of AI use on occupational health equity (biased outcomes). OSH researchers and practitioners need to advocate for a long-term strategy in partnership with government, AI experts, and industry for protecting the health, safety, and well-being of all workers.

Application of REDECA Framework to Improve the Safety and Health of Agricultural Tractor Drivers

This chapter describes a framework to review the application of AI in OSH in main industry sectors. The chapter is under review.

1.21. Introduction

Agriculture has the highest rates of fatality incidents in the US [101]. While fatal incidents dropped from about 1000 cases in the early 1990s to less than 600 cases in 2019, this was mostly due to fewer workers and more efficient machinery [102]. Moreover, within all fatal agricultural injury types (tractor, roadway, grain bins, farm equipment, All Terrain Vehicle (ATV), electrocution, animals, manure storage, etc.), the numbers of the tractor-related injuries remain high with 213 cases from 1999 to 2019 [96], [101].

Over the past few years, several efforts have been made to improve the safety of agricultural workers. These include research, teaching, and extension. In the case of tractor related injuries for instance, the focus has been on improving the design and functionality of tractors; while the teaching and extension focus on providing farmers safe practices. Despite such efforts, fatal tractor related incidents are common. Thus, the need to explore different approaches to tackle tractor related injuries.

AI has been applied to many domains. In 2019 alone, over 20,000 papers were published to show the application of AI in various industries [78]. Both academia and industries have used AI to address a variety of issues, including decision making [82], environmental monitoring [83], [84], operational cost reduction [85], and productivity [86]. Machine learning algorithms have also been used to detect vocal disorders in workers who frequently use their voices [89], and to detect indoor crews in the event of fire. Gomez-Gil et al. used EMG readings to steer a tractor with almost the same accuracy as manual steering [103]. Szczepaniak et al. developed models to assess the stability and steerability of agricultural machines that could be adapted to drivers' characteristics to improve safety [104]. Sensors can measure vibrations experienced by farmers using agricultural aircraft. Tri-axial accelerometers were used to measure acceleration at the seat level [105]. Kociolek et al. showed that operators on quad bikes were exposed to head and neck vibration higher than the permissible levels of exposure [106]. Similarly, Calvo et al. used three different accelerometers to measure hand-to-arm vibration and repetitive action (OCRA) levels for farmers who used power tillers. Result indicated vibrational exposure far above acceptable exposure levels [107]. These studies highlight mounting evidence that AI technology can successfully detect, identify, and forecast unsafe behavior in potentially dangerous working environments.

REDECA [77] is a novel framework developed by the authors to theorize how AI methods can anticipate and control risk of exposure in a worker's immediate environment. The REDECA framework is based on the Swiss cheese model [95] that depicts how incidents of injury occur in complex systems. This model can conceptualize multiple layers of defenses and safeguards (or interventions) to prevent incidents. The REDECA framework includes several elements. First, the different states workers can be. R1 is where workers have minimal to no risk of exposure. R2 indicates exposure to hazard and increased risk of injury. R3 indicates a harmful work-related event. Second, are monitoring transitions to adverse states. This can be technologies that predict the probability of transitioning among states or technologies that detect transitions among states. Finally, are intervention strategies to keep workers safe or reduce the impact of an adverse event.

The objectives of this work are to: (1) identify root causes of agricultural tractor driver incidents; (2) apply the REDECA framework to determine all steps/ stages involved before and after occurrence of incidents (3) determine existing AI solutions to reduce the agricultural tractor driver incidents; (4) identifying opportunities for both industry and academia to propose new AI interventions within missing REDECA elements to improve the safety of tractor drivers in the future; and (5) provide good general practices to improve the safety of tractor drivers.

1.22. Materials and Methods

Agriculture remains the most dangerous occupation in the U.S. [101]. Among all agricultural related injuries, tractor related injuries are highest [96], [101]. Several databases track all occupational incidents such as Ag Injury News Clippings, Fatality Assessment and Control Evaluation, and bureau of labor statistics. However, the Fatality Assessment and Control Evaluation (FACE) is the most complete and comprehensive. We thus studied FACE reports, from the Center for Disease Control and Prevention (CDC) website, related to fatal incidences among agricultural tractors drivers.

We used the following procedures to extract and analyze FACE reports data. (1) We accessed FACE report data at national and state levels from the CDC website, (2) using the keywords “agriculture” to find agricultural related reports, (3) and “machine farming” to find the machine related cases, (4) then extracted and saved the reports to an excel file. (5) We searched the excel file for the keyword “tractor” to identify all tractor related cases. (6) In cases where national and state reports were identical, we analyzed the cases as a single incident. All reports were reviewed and unwitnessed reports were excluded from analysis.

Further analysis (7) categorized reports based on types of tractor related incident: roll over, run over, overturn, pinned by, fall, others that include fire, crashed incidents as well as other types of incidents. Additionally, (8) in each category, reports with the same causes were sub-categorized together. Categories and subcategories are detailed in Table 7-1.

Table 7-1: Incident categories and sub-categories.

Types of Categories	Types of Sub- Categories
Run Over	(1) Run over due to unsecured seat placement. (2) Run over due to the tractor left in gear after its last use. (3) Run over because of the weight of the water tank trailer and high speed. (4) Run over because the tractor engine was operating at high rpm, slipped into gear, and accelerated forward at high speed. (5) Run over due to the wheels on the right side of the tractor running over a tree stump. (6) Run over due to the tractor being in forward gear and the driver standing on the ground trying to turn the tractor on. (7) Run over due to tractor being left on a hill while running without driver and dismounted from its initial attached loader bucket. (8) Run over due to driver having a seizure and falling out off the tractor because they did not use a seat belt. (9) Run over due to unintended door opening causing the driver to fall out of the tractor. (10) Run over due to non-functional brake. (11) Run over due to non-functional gear. (12) Run over due to the driver falling off from the tractor moving at high speed. (13) Run over due to a driver falling off the tractor while moving. The driver tried to jump on the tractor, slipped and engaged the gear making the tractor move while the victim was on the ground. (14) Run over due to falling off the tractor while making a sharp turn. (15) Run over due to poor driver visibility.
Pinned by	(1) Pinned between the enclosed auger and the tractor steering wheel due to an accident between the tractor and barn wall. (2) Pinned to tree while inside tractor cab. (3) Pinned by a truck bed attached to a tree by a chain. (4) Pinned by and compressed by an attachment while trying to detach it.

Types of Categories	Types of Sub- Categories
	(5) Pinned by a tractor that drove into a ditch where the victim was located. (6) Pinned by hayrack roll over that pinned the victim to the left rear tractor tire.
Fall	(1) Fall out of the tractor while driving due to an unattached seat. (2) Fall off due ti slip on a tractor step. (3) Fall off due to tractor driving at high speed on a incline and driver not wearing a seatbelt. (4) Fall off from tractor due to heart issues and not wearing a seatbelt. (5) Fall off due to tractor jumping jack-knifed causing the driver to be thrown off from the tractor. (6) Fall off from tractor due to loss of control while driving.
Other category including fires and crashes	(1) Victim struck by hay bale that fell out of elevated bucket. (2) Victim fell while re-attaching a shaft to a tractor due to control lost in the hollow. (3) Entrapment due to victim's shirt entrapped inside an auger not equipped with guard. (4) Entanglement due tractor left on and hay baler left on downward slope. (5) Incident due to a tractor driver hauling a tree higher than the recommended height. (6) Crash due to a semi-truck hitting a tractor causing both victims to be ejected from their respective vehicles. (7) Crash due to a detached wagon moving forward, crushing the user against the tractor. (8) Fire due to the victim puncturing an above ground gas line with a tractor due to poor visibility. (9) Fire due to vinyl shrouds bursting into flames, spreading to the victim's clothing. (10) Fire due to ignition of a tractor struck by a tree.
Roll Over	1) Roll over due to a tractor's sharp turn into a ditch. 2) Roll over due to tractor driving and reaching a steep roadside.

Types of Categories	Types of Sub- Categories
	3) Roll over due de-attaching a wagon from the tractor on a hilly, sharply curved road. 4) Roll over due to an attached overweight hay bale and driving on a slope. 5) Roll over due to tractor driving on an irregular surface causing its center of gravity to shift resulting in tractor instability. 6) Roll over due to a rasied tractor bucket, and attempting to make a left turn while on the incline. 7) Roll over due to driver foot slipping off the tractor's clutch while trying to pull out another tractor stuck in mud. 8) Roll over due to the tractor being stopped and dismounted without setting the brakes and leaving the manual transmission in gear. 9) Roll over due to the tractor not equipped with Roll-Over Protective Structure (ROPS) and seat belt reaching an embankment while the brakes were not engaged.
Overturn	1) Overturn due to wet/snow/muddy hill causing the tractor to slide down. 2) Overturn due to tractor driving on slope terrain and ditch. 3) Overturn due to tractor driving on the right side of the road with a heavy trailer was connected and attempting to overcorrect. 4) Overturn due to the tractor turning on a slope/hill. 5) Overturn due to the tractor sliding down a steep embankment. 6) Overturn due to the tractor attached to overweight objects. 7) Overturn due to the tractor wheel coming off the ground while driving. 8) Overturn due non-functional gear shift. 9) Overturn due tractor trying to pull out another tractor stuck into mud. 10) Overturn due to speeding tractor.

We identified each state of work using the REDECA framework – from when tractor drivers started working until the hazard happened (R1, R2, R3). We then indicated AI technologies that could predict the probability of transitioning into next states, detect transitions among states, and indicate intervention strategies to keep the tractor drivers safe or reduce recovery times when hazard happened. Finally, we found existing AI technologies used by other industries and advised they be used in the agriculture industry to improve the safety and health of tractor drivers. Figure 7-1 gives an overall view of our methodology.

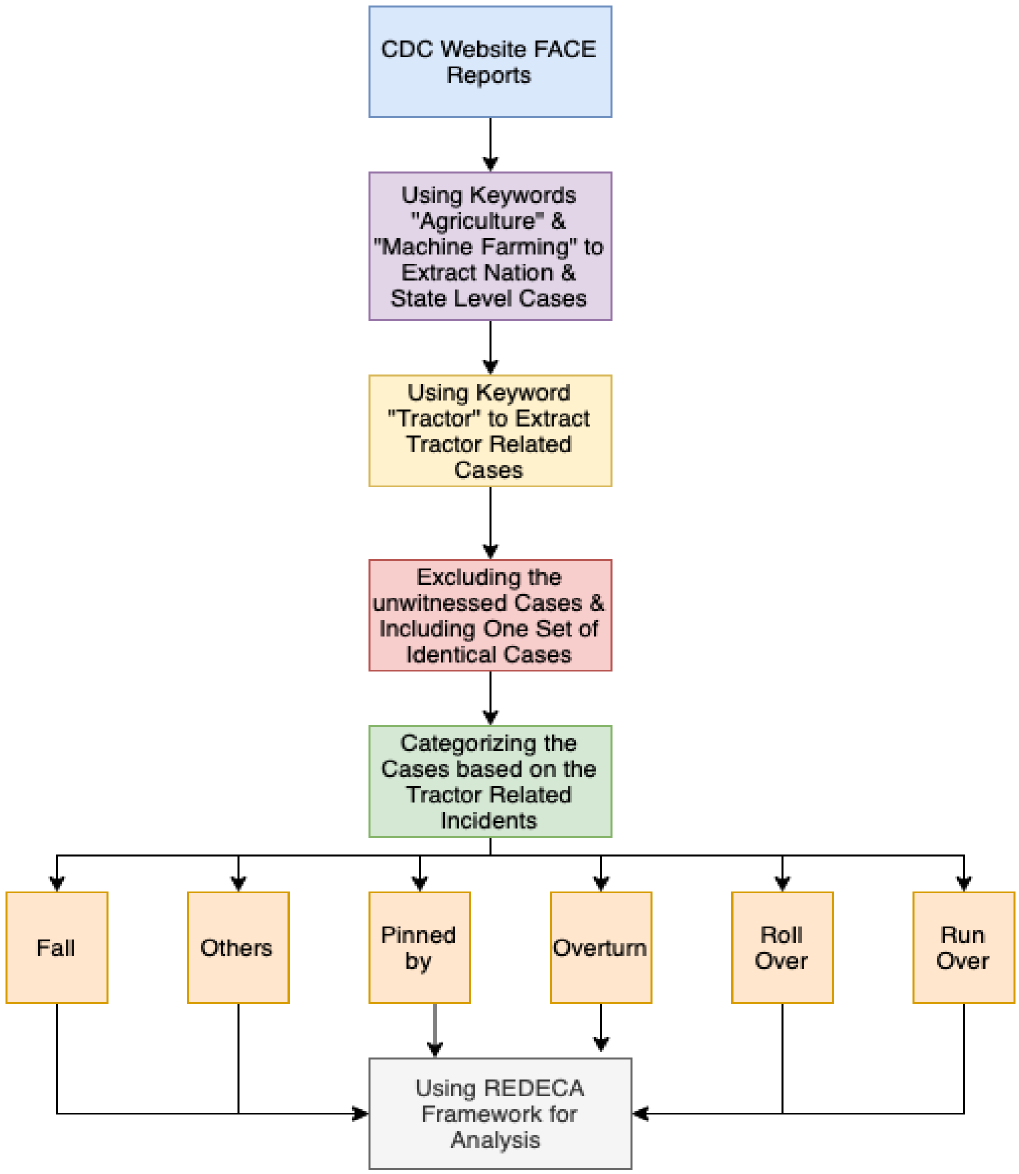

Figure 7-1: The overall view of the methodology

1.23. Results

From 442 initial cases of fatalities from FACE reports at nation and state levels, we identified 188 tractor related incidents using the keyword "tractor." of these, 9 cases were unwitnessed and 16 were duplicates that we combined into singles cases. This left 171 cases for analysis.

We sub–categorized 160 cases by the kind of tractor-related occurrence, yielding 52 cases of run over, 44 cases of roll over, 7 of pinned by, 42 cases of overturned, 10 cases of fall, and 16 cases of others, including fires, and crashes. We further sub–categorized incidents with similar causes: run over had 15 sub-categories, pinned by and fall both had 6, others and overturn both had 10, and roll over had 9.

We analyzed all cases using the REDECA framework developed in [77]. Table 7-1 summarizes Appendix A from [108]. In the case of run over for example, from a review of 52 cases we obtained 15 sub-categories, each with different occurrences. There was only one report of classes 1, 3, 5, 8, 9, 13, and 15 respectively. There were two reports of classes 2, 4, 7, 11, and 14 respectively. Finally, there were 21 reports of class 6, 3 reports of class 10, and 4 reports for class 12 respectively.

Based on [108], Table A1, 15 AI solutions exist to measure the probability of entering R3, 2 AI solutions to detect change between R2 to R3, 11 AI interventions to prevent entry to R3, and 4 AI interventions to minimize damage and recovery in R3. However, AI technology cannot establish the probability of reducing recovery time in R3 and detecting transitions between R2 and R3, and Intervention to send workers to R2.

In the case of pinned, we created 6 subcategories from the 6 cases we reviewed. Each class was reported 1 time respectively. Based on [108], Table A2, 6 AI solutions exist to measure the probability of entering R3. However, there are no AI solutions for the probability of reducing recovery time for R3, detecting change between R2 to R3, intervention to prevent entry to R3, Intervention to send workers to R2, and intervention to minimize damage and recovery in R3. There are 6 AI interventions to prevent entry to R3.

In case of fall, we created 6 classifications from the 6 cases reviewed, with 1 report per case except one classification with 2 reports. Based on [108], Table A3, there are 6 existing AI solutions to measure the probability of entering R3. However, are no AI solutions for the probability of reducing recovery time for R3, detecting change between stages R2 to R3, Intervention to send workers to R2, and intervention to prevent entry to R3. There are 2 AI interventions to prevent entry to R3. Finally, there is only 1 AI intervention to minimize damage and recovery.

In cases of "others," we created 10 classifications from the 10 cases we reviewed, each with 1 reported case. Based on [108], Tables A.4 and A.6, 10 AI solutions exist to measure the probability of entering R3. However, there are no AI solutions for the probability of reducing recovery time in R3, detecting change between R2 to R3, Intervention to prevent entry to R3, and intervention to send workers to R2. There are 7 AI interventions to prevent entry to R3. Finally, there are no AI interventions to send workers to R1 and to minimize damage, but there are 2 AI interventions to minimize damage and recovery.

In cases of roll over, we obtained 9 classifications from the 44 cases we reviewed. Each classification had different numbers of reports: class 1, 3, and 5 contain 6 reports respectively. Class 4, 6, and 9 contain 3 reports respectively. Finally, class 2, 7, and 8 include 4, 1, and 2 reports respectively.

Based on [108], Table A5, only 5 AI solutions exist to measure the probability of entering R3 and 5 AI interventions to prevent entry to R3. Except where the AI solutions are not applicable within the context of a tractor driver, there are no existing AI solutions applicable to all scenarios of the REDECA framework. There are no AI solutions for the probability of reducing recovery time in R3, detecting change between R2 to R3, intervention to prevent entry to R3, intervention to send workers to R2, and intervention to minimize damage and recovery in R3.

In cases of overturn, we created 10 classifications from the 42 cases we reviewed. Each classification had different numbers of reports: class 3, 7, 8, 9, and 10 each reported 1 time respectively. Class 1, 2, 4, 5, and 6 had 10, 12, 4, 11, and 6 reports respectively. Based on [108], Table A6, only 5 AI solutions exist to measure the probability of entering R3 and 5 AI Interventions to prevent entry to R1. Except where an AI solution is not applicable within the context of a tractor driver, there are no existing AI solutions applicable to all scenarios of the REDECA framework. There are no AI solutions for the probability of reducing recovery time in R3, detecting change between R2 to R3, intervention to prevent entry to R3, intervention to send workers to R2, and intervention to minimize damage and recovery in R3.

1.24. Discussion

The application of AI to several industrial domains has been described as the Fourth Industrial Revolution [97]. Innovations in artificial intelligence using sensors, robots, and ML algorithms have been shown to increase productivity and improve the safety and health of workers in the workplace. In agriculture, despite tremendous efforts to improve safety, the number of the tractor related incidents remains high [96], [101]. REDECA is a state-of-the-art framework that identifies work states from the start of the workflow until a hazard occurs (R1, R2, R3). It then categorizes AI use in OSH to highlight strengths, opportunities, and weaknesses clearly and efficiently.

In this work, we extracted tractor–related fatality incidents from FACE reports on the CDC website. We then used the REDECA framework to assess processes before incidents and identified potential AI solutions that can reduce these incidents.

For run over cases, class 6, that is "driver run over by tractor in gear, while driver was standing on ground starting the tractor, and delay in receiving care," had the most reports. [108]. Such cases are therefore critical and AI solutions should be developed to reduce their occurrence. Moreover, our REDECA framework analysis of run over cases, highlighted a lack of AI technology to reduce the probability of recovery time in R3, to detect changes between R2 and R3, and interventions to send workers to R2. As a result, researchers and industry have an opportunity to address said elements.

Also, after analyzing FACE reports, we have come up with the following recommendations to reduce run over incidents, and consequently improve the health and safety of tractor drivers. 1) Sensors alerting the driver to turn off the tractor before leaving the tractor. 2) Sensors alerting the driver not to turn on the tractor while on the ground. 3) Smart technology to keep the tractor in a parked state while in a maintenance state. 4) Pressure sensors to inform the driver not to leave the seat while tractor is running. 5) Sensor preventing the tractor from starting while not in neutral. 6) Operators of tractors that are equipped with rollover protective structure and a seat belt engaged while operating the tractor. 7) Tractor transmissions should always be put in park before dismounting to hook up equipment or adjust.

In brief, for pinned cases, there is a lack of AI technology to detect change between R2 to R3, to minimize damage and recovery in R3, to reduce the probability of recovery time for R3, to prevent entry to R3, and to send workers to R2. Hence, researchers and industries can focus on developing missing AI technology that improves the safety of tractor drivers.

We recommend the following to reduce pinned by incidents: 1) Add roughness to smooth clutch or brake pedals by using a 4-inch portable grinder or welding a bead of metal on the pedal or cover the pedal with a non-slip surface for added foot pedal control. 2) Carrying a reliable 2-

way communication device for emergency communication in case of injury and emergency situations.

In brief, for fall cases, there is a lack of AI technology to detect change between R2 to R3, to minimize damage and recovery in R3, to reduce the probability of recovery time for R3, to prevent entry to R3, and to send workers to R2. Thus, there is a need for academic and industrial research to develop these nonexistent AI technologies.

To reduce fall incidents, we suggest: 1) Owners/operators of tractors should ensure that the tractor seat is in good condition and firmly attached to the base. (Vibration sensors or any sensors could alert the user when seats are not properly attached to the base or raised) 2) Tractor manufacturers should give more attention to the safe design of steps and handrails to further increase operator safety. 3) Tractor operators should pay close attention to symptoms of illness and should seek prompt medical attention.

For other cases including fire and crashes, there is a lack of AI technology to detect change between R2 to R3, to reduce the probability of recovery time for R3, to prevent entry to R3, and to send workers to R2. Therefore, developing new AI technology is crucial for these missing elements of the REDECA framework.

To reduce pinned by incidents, we suggest the following: 1) Ensure adequate rest and minimize distractions while driving. 2) Use less busy alternate routes when available when operating agricultural equipment on the road, especially during high traffic volume hours. 3) Install side view mirrors and construct/purchase appropriate temporary flashing warning lights and attach them to a tractor if not so equipped when the tractor is operated on the road. 4) MIFACE recommends that the tractor PTO stub shaft be guarded at all times to prevent entanglement. 5) Operators should not wear loose fitting clothing when operating farm machinery. Clothing manufacturers should therefore consider developing work clothes that tear away in case of entanglement and label clothing when tear resistant.

For roll over cases, class 1 is “roll over due to tractor sharp turning into a ditch. No ROPS and seat belt”, class 3 which is “roll over” due de-attaching the wagon from the tractor in a hilly road and sharp curve road. No ROPS and seat belt”, and class 5 which is “roll over due to tractor driving on an irregular surface causing its center of gravity to shift which resulting in tractor instability” has the highest numbers of six reports. This implies such cases are critical and AI solutions should be developed to reduce their occurrence.

Moreover, based on the REDECA framework analysis for roll over cases, there is a lack of AI technology for the probability of reducing recovery time in R3, detecting changes between R2 and R3, and intervention to send workers to R2, intervention to prevent entry to R3, and intervention to minimize damage and recovery in R3. Consequently, new AI technologies are welcome to address the needs.

Also, after analyzing the reports, we have come up with the following recommendations to reduce the roll over incidents to improve the health and safety of the tractor driver. 1) Foreign farm laborers should be trained in their native language on safe operation of farm machinery and informed of dangers of local terrain. 2) Tractor front end loader operators should be made aware of overturn hazard and methods to reduce this hazard, including safe driving on sloping grounds, the changing center of gravity caused by a loader bucket, keeping the bucket low while driving, and using counterweights on the tractor. 3) Be aware of the dangers of fatigue and weariness when operating tractors and take frequent breaks.

In brief, for overturn cases, class 1 which is “overturn due to wet/snow/muddy hill causing the tractor to slide down. No ROPS or seatbelt”, class 2 “overturn due to tractor driving on slope terrain and ditch. No ROPS or seatbelt”, and class 5 “overturn due to tractor sliding down a steep embankment” contain 10, 12, and 11 reports respectively. Thereafter, given the high negative impact of these cases, there is an urgent need to develop AI solutions.

Moreover, based on the REDECA framework analysis for roll over cases, there is a lack of AI technology for the probability of reducing recovery time in R3, detecting changes between R2 and R3, and intervention to send workers to R2, intervention to prevent entry to R3, and intervention to minimize damage and recovery in R3. So, new AI technologies are needed to fulfill the missing REDECA framework elements.

Also, after analyzing the reports, we have come up with the following recommendations to reduce the roll over incidents to improve the health and safety of the tractor driver. 1) Tractor operators should be trained to recognize and understand the hazards associated with towing items that exceed the weight of the tractor. 2) When driving a tractor on a public road, the driver should maintain a safe and well-defined position on the road in the correct traffic lane. The tractor operator should not pull off the road to allow traffic to pass unless there is a safe and stable location to maneuver. 3) Provide personal communication devices to workers assigned to remote worksites. 4) Front-end counterweights should be used to improve traction and stability.

In addition to all other specific recommendations provided above for each type of incidents cases, we observed that some causes of incidents were common across all these types of incidents. That being said, the following recommendations would help address all these common causes of incidents: 1) Tractor drivers should always use the seat belts, and tractors should be equipped with ROPS. 2) Have a trained mechanic inspect used equipment prior to use to ensure equipment has all safety features intact and to note any equipment modifications that may affect equipment performance and function. 3) Routinely inspect tractors to identify potential safety issues, such as old/faded SMV emblems, missing PTO master shield, and ROPS availability. Install/re-install missing or damaged items. 4) Ensure medical conditions are managed by all workers on the farm. 5) Tractor operators should maintain safe operating speeds at all times. 6) Survey the work site to identify hazards. All employees should then be informed of the possible hazards and encouraged to report any unsafe work conditions. 7) Working youth should only be assigned age appropriate tasks. 8) Farmers, rural residents, and county/state road departments should pursue grading changes or post warning signs along the roadway to alert drivers of dangerous intersections with a farm lane or driveway. 9) Operators should lock both brake pedals together before driving in slippery conditions.

PROPOSAL AND FUTURE WORK

1.25. CONCLUSION

This dissertation first emphasizes a process mining pre-process step which concatenates some events which holds concurrency relationship based on a probability-based function. Further this dissertation demonstrates a process mining/ deep learning approach to predict three different healthcare outcomes which are unplanned 30-day readmission of the ICU patients with HF, mortality of the COVID patients every 6-hours within 72 hours of the patients' admission to the hospital, and mortality of ICU patients with PI. Moreover, the dissertation focuses on applying the pre-processing algorithm on the healthcare event logs to evaluate its effectiveness in this domain. Additionally, this dissertation introduces REDECA framework to review the application of AI in OSH in main industries and identifies gaps which can be fulfilled with AI to improve the Safety of the worker. At the end, REDECA framework is applied specifically on agricultural tractor drivers. The contribution of this dissertation can be summarized as follows and is obtained from [19], [32]–[34], [67], [77]:

Improving process discovery algorithm using event concatenation: in this contribution a pre-processing algorithm is developed which concatenates events holding concurrent relations. As a result, a higher quality process model is generated by process discovery algorithms. The resultant process models are much simpler and have higher F-measure compared to using the raw data for process discovery. The evaluation metrics confirms significant after applying the pre-processing step.

Prediction of critical health outcomes using process mining/ deep learning approach: this contribution is focused on an approach to accurately predict the important health outcomes by combining the process mining and deep learning techniques. The approach first generates a process model through a process discovery algorithm. The process model along with the event logs were then fed to DREAM algorithm to produce the TSS. The resultant TSS, demographic information of the patients, and the severity scores related to each patient are then fed to a NN to train a deep learning model for prediction. These methods outperformed current state-of-the-art methods on three case studies.

Effect of process mining pre-processing step on the prediction of the critical health outcomes: this contribution demonstrates the effectives of the pre-processing on real-life healthcare datasets. EHR are converted to the event logs and fed to the concatenation algorithm. The resultant event logs are then fed to the process discovery algorithm to produce the process model. The process model is then evaluated by using common evaluation metrics. Moreover, the resultant process model, the event logs, demographics of the patients, and severity scores are fed to the DREAM algorithm to predict the critical health outcomes and the prediction model is evaluated through AUC metric. The same procedure is tried by using raw event log and the results are compared. The method shows significant improvements.

REDECA: A Novel Framework to Review Artificial Intelligence and Its Applications in Occupational Safety and Health. A new framework, REDECA, is introduced to identify gaps in the application of AI in OSH. REDECA then bridges these gaps to improve worker health and safety in agriculture, oil and gas, mining, transportation, and construction industries. This method is useful as a general framework to improve the safety and health of the workers in workplaces in multiple industries.

Application of REDECA Framework to Improve the Safety and Health of the Agricultural Tractor Drivers: 171 Fatality FACE reports of tractor drivers are reviewed and categorized into six main categories. The REDECA framework is used to review existing AI solutions, to identify root causes of agricultural tractor driver incidents, and to suggest existing AI solutions to reduce incidents, thus improve tractor driver safety and health.

1.26. FUTURE WORK

This dissertation can be advanced in several ways by future works:

Concatenation is done only on events which hold concurrency. However, other event relationships such as choice and sequential exist. Developing new pre-processing algorithms that consider other event relationships and compare the results to the original would be a next step. The p^* threshold is found manually with the steps of 0.1, therefore developing a method to find optimal p^* would be an important next step.

The DREAM algorithm used to predict several critical health outcomes in this dissertation, basic information from EHRs such as admission and insurance type, lab measurements, and comorbidities are converted to the event logs. However, while more advanced information such as clinical notes and medical images are available, they are not included because they are inherently difficult to convert to suitable inputs to the process mining/ deep learning framework. Expanding the framework to include such data to improve the predictive capability of the model would be considered.

The REDECA framework can be used as a check list in any workplaces to find where AI in any work state (R1, R2, R3) is applied, or not, to improve safety and health of the workers. Thus, choosing workplaces in hazardous industries where AI could play a potential role improving worker safety and protecting their lives could be a next step. In this way the lack of AI would be identified in workplaces, thus, new AI interventions, algorithms, and automations that improve the safety and the health of the workers could be proposed for future work.

CITED LITERATURE

[1] W. van der Aalst, *Process mining: Data science in action*. 2016. doi: 10.1007/978-3-662-49851-4.

[2] J. Theis and H. Darabi, "Decay Replay Mining to Predict Next Process Events," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2937085.

[3] H. Darabi, W. L. Galanter, J. Y. Y. Lin, U. Buy, and R. Sampath, "Modeling and integration of hospital information systems with petri nets," in *2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, SOLI 2009*, 2009. doi: 10.1109/SOLI.2009.5203928.

[4] S. Suriadi, M. T. Wynn, C. Ouyang, A. H. M. ter Hofstede, and N. J. van Dijk, "Understanding process behaviours in a large insurance company in Australia: A case study," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7908 LNCS. doi: 10.1007/978-3-642-38709-8_29.

[5] N. Wightkin, U. Buy, and H. Darabi, "Formal modeling of sequential function charts with time petri nets," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, 2011, doi: 10.1109/TCST.2010.2047106.

[6] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, vol. 136, no. 2. 2011.

[7] V. A. Rubin, A. A. Mitsyuk, I. A. Lomazova, and W. M. P. van der Aalst, "Process mining can be applied to software too!," in *International Symposium on Empirical Software Engineering and Measurement*, 2014. doi: 10.1145/2652524.2652583.

[8] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans Knowl Data Eng*, vol. 16, no. 9, 2004, doi: 10.1109/TKDE.2004.47.

[9] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible heuristics miner (FHM)," in *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*, 2011. doi: 10.1109/CIDM.2011.5949453.

[10] S. K. L. M. vanden Broucke and J. de Weerdt, "Fodina: A robust and flexible heuristic process discovery technique," *Decis Support Syst*, vol. 100, 2017, doi: 10.1016/j.dss.2017.04.005.

[11] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "On the role of fitness, precision, generalization and simplicity in process discovery," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7565 LNCS, no. PART 1. doi: 10.1007/978-3-642-33606-5_19.

[12] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs - A constructive approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7927 LNCS. doi: 10.1007/978-3-642-38697-8_17.

[13] A. Augusto, R. Conforti, M. Dumas, M. la Rosa, and A. Polyvyanyy, "Split miner: automated discovery of accurate and simple business process models from event logs," *Knowl Inf Syst*, vol. 59, no. 2, 2019, doi: 10.1007/s10115-018-1214-x.

[14] W. Bose, R P Jagadeesh Chandra, Van der Aalst, "Analysis of Patient Treatment Procedures: The BPI Challenge Case Study," in *Business Process Management Workshops (1) (Vol. 99, pp. 165-166).*, 2011.

[15] W. van der Aalst *et al.*, "Process Mining Manifesto," 2012, pp. 169–194. doi: 10.1007/978-3-642-28108-2_19.

[16] S. Rogers, "Big Data is Scaling BI and Analytics," New York, Sep. 2011.

[17] B. van Dongen, "BPI Challenge 2012," *BPI Challenge 2012*, 2012. https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204 (accessed Oct. 04, 2022).

[18] B. F. and van der A. W. M. P. Buijs Joos C. A. M. and van Dongen, "On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery," in *On the Move to Meaningful Internet Systems: OTM 2012*, 2012, pp. 305–322.

[19] M. Pishgar, M. Razo, and H. Darabi, "Improving Process Discovery Algorithms Using Event Concatenation," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3185235.

[20] L. Ghionna, G. Greco, A. Guzzo, and L. Pontieri, "Outlier detection techniques for process mining applications," in *SEBD 2008 - Proceedings of the 16th Italian Symposium on Advanced Database Systems*, 2008.

[21] NIH, "Markov Clustering (MCL): a cluster algorithm for graphs," *Biowulf: High Performance Computing at the NIH*. https://hpc.nih.gov/apps/mcl.html (accessed Oct. 04, 2022).

[22] X. Lu, D. Fahland, F. J. H. M. van den Biggelaar, and W. M. P. van der Aalst, "Detecting deviating behaviors without models," in *Lecture Notes in Business Information Processing*, 2016, vol. 256. doi: 10.1007/978-3-319-42887-1_11.

[23] C. W. Günther and W. M. P. van der Aalst, "Fuzzy mining - Adaptive process simplification based on multi-perspective metrics," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4714 LNCS. doi: 10.1007/978-3-540-75183-0_24.

[24] H. J. Cheng and A. Kumar, "Process mining on noisy logs - Can log sanitization help to improve performance?," *Decis Support Syst*, vol. 79, 2015, doi: 10.1016/j.dss.2015.08.003.

[25] R. Conforti, M. la Rosa, and A. H. M. ter Hofstede, "Filtering Out Infrequent Behavior from Business Process Event Logs," *IEEE Trans Knowl Data Eng*, vol. 29, no. 2, 2017, doi: 10.1109/TKDE.2016.2614680.

[26] M. Fani Sani, S. J. van Zelst, and W. M. P. van der Aalst, "Applying sequence mining for outlier detection in process mining," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11230 LNCS. doi: 10.1007/978-3-030-02671-4_6.

[27] S. Suriadi, R. Andrews, A. H. M. ter Hofstede, and M. T. Wynn, "Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs," *Inf Syst*, vol. 64, 2017, doi: 10.1016/j.is.2016.07.011.

[28] N. Tax, N. Sidorova, and W. M. P. van der Aalst, "Discovering more precise process models from event logs by filtering out chaotic activities," *J Intell Inf Syst*, vol. 52, no. 1, 2019, doi: 10.1007/s10844-018-0507-6.

[29] 4TU.ResearchData, "4TU.ResearchData," 2021. https://data.4tu.nl/ (accessed Oct. 04, 2022).

[30] W. M. P. van der Aalst, A. Bolt, and S. J. van Zelst, "RapidProM: Mine Your Processes and Not Just Your Data," Mar. 2017.

[31] R. Conforti, M. Dumas, L. García-Bañuelos, and M. la Rosa, "BPMN Miner: Automated discovery of BPMN process models with hierarchical structure," *Inf Syst*, vol. 56, 2016, doi: 10.1016/j.is.2015.07.004.

[32] M. Pishgar, J. Theis, M. del Rios, A. Ardati, H. Anahideh, and H. Darabi, "Prediction of unplanned 30-day readmission for ICU patients with heart failure," *BMC Med Inform Decis Mak*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12911-022-01857-y.

[33] M. Pishgar *et al.*, "A process mining- deep learning approach to predict survival in a cohort of hospitalized COVID-19 patients," in *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, BioMed Central Ltd, 2022. doi: 10.1186/s12911-022-01934-2.

[34] M. Pishgar, M. Razo, J. Theis, and H. Darabi, "Process Mining Model to Predict Mortality in Paralytic Ileus Patients," in *2021 International Conference on Cyber-Physical Social Intelligence, ICCSI 2021*, 2021. doi: 10.1109/ICCSI53130.2021.9736217.

[35] D. Mozaffarian *et al.*, "Heart Disease and Stroke Statistics-2016 Update A Report From the American Heart Association," 2015. [Online]. Available: http://my.americanheart.org/statements

[36] S. B. Golas *et al.*, "A machine learning model to predict the risk of 30-day readmissions in patients with heart failure: A retrospective analysis of electronic medical records data," *BMC Med Inform Decis Mak*, vol. 18, no. 1, 2018, doi: 10.1186/s12911-018-0620-z.

[37] B. Y. Tan, J. Y. Gu, H. Y. Wei, L. Chen, S. L. Yan, and N. Deng, "Electronic medical record-based model to predict the risk of 90-day readmission for patients with heart failure," *BMC Med Inform Decis Mak*, vol. 19, no. 1, 2019, doi: 10.1186/s12911-019-0915-8.

[38] C. R. Ponzoni *et al.*, "Readmission to the intensive care unit: Incidence, risk factors, resource use, and outcomes: A retrospective cohort study," *Ann Am Thorac Soc*, vol. 14, no. 8, 2017, doi: 10.1513/AnnalsATS.201611-851OC.

[39] D. E. Singer, A. G. Mulley, G. E. Thibault, and G. O. Barnett, "Unexpected Readmissions to the Coronary-Care Unit during Recovery from Acute Myocardial Infarction," *New England Journal of Medicine*, vol. 304, no. 11, 1981, doi: 10.1056/nejm198103123041101.

[40] T. Desautels *et al.*, "Prediction of early unplanned intensive care unit readmission in a UK tertiary care hospital: A cross-sectional machine learning approach," *BMJ Open*, vol. 7, no. 9, 2017, doi: 10.1136/bmjopen-2017-017199.

[41] A. Ashfaq, A. Sant'Anna, M. Lingman, and S. Nowaczyk, "Readmission prediction using deep learning on electronic health records," *J Biomed Inform*, vol. 97, 2019, doi: 10.1016/j.jbi.2019.103256.

[42] Z. Hu and D. Du, "A new analytical framework for missing data imputation and classification with uncertainty: Missing data imputation and heart failure readmission prediction," *PLoS One*, vol. 15, no. 9 September, 2020, doi: 10.1371/journal.pone.0237724.

[43] J. M. Vader *et al.*, "Timing and Causes of Readmission After Acute Heart Failure Hospitalization—Insights From the Heart Failure Network Trials," *J Card Fail*, vol. 22, no. 11, 2016, doi: 10.1016/j.cardfail.2016.04.014.

[44] Y. W. Lin, Y. Zhou, F. Faghri, M. J. Shaw, and R. H. Campbell, “Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long shortterm memory,” *PLoS One*, vol. 14, no. 7, 2019, doi: 10.1371/journal.pone.0218942.
[45] M. Ghasemi and D. Amyot, “Process mining in healthcare: A systematised literature review,” *International Journal of Electronic Healthcare*, vol. 9, no. 1. 2016. doi: 10.1504/IJEH.2016.078745.
[46] P. Baruah, “Predicting Hospital Readmission using Unstructured Clinical Note Data,” 2020.
[47] J. Theis, W. L. Galanter, A. D. Boyd, and H. Darabi, “Improving the In-Hospital Mortality Prediction of Diabetes ICU Patients Using a Process Mining/Deep Learning Architecture,” *IEEE J Biomed Health Inform*, vol. 26, no. 1, 2022, doi: 10.1109/JBHI.2021.3092969.
[48] A. Johnson, T. Pollard, and R. Mark, “MIMIC-III Clinical Database (version 1.4),” *PhysioNet*, 2016.
[49] A. Elixhauser, C. Steiner, D. R. Harris, and R. M. Coffey, “Comorbidity Measures for Use with Administrative Data,” *Med Care*, vol. 36, no. 1, 1998, doi: 10.1097/00005650-199801000-00004.
[50] S. E. S. Brown, S. J. Ratcliffe, and S. D. Halpern, “An empirical derivation of the optimal time interval for defining ICU readmissions,” *Med Care*, vol. 51, no. 8, 2013, doi: 10.1097/MLR.0b013e318293c2fa.
[51] M. E. Charlson, P. Pompei, K. L. Ales, and C. R. MacKenzie, “A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation,” *J Chronic Dis*, vol. 40, no. 5, 1987, doi: 10.1016/0021-9681(87)90171-8.
[52] E. R. Delong, D. M. Delong, and D. L. Clarke-Pearson, “Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach,” 1988. [Online]. Available: https://about.jstor.org/terms
[53] S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-December.
[54] X. Liu, Y. Chen, J. Bae, H. Li, J. Johnston, and T. Sanger, “Predicting Heart Failure Readmission from Clinical Notes Using Deep Learning,” in *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*, 2019. doi: 10.1109/BIBM47256.2019.8983095.
[55] K. Huang, J. Altosaar, and R. Ranganath, “ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission,” Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.05342
[56] M. Feng *et al.*, “Transthoracic echocardiography and mortality in sepsis: analysis of the MIMIC-III database,” *Intensive Care Med*, vol. 44, no. 6, 2018, doi: 10.1007/s00134-018-5208-7.
[57] A. P. Kurniati, E. Rojas, D. Hogg, G. Hall, and O. A. Johnson, “The assessment of data quality issues for process mining in healthcare using Medical Information Mart for Intensive Care III, a freely available e-health record database,” *Health Informatics J*, vol. 25, no. 4, 2019, doi: 10.1177/1460458218810760.
[58] B. S. Wessler *et al.*, “External Validations of Cardiovascular Clinical Prediction Models: A Large-Scale Review of the Literature,” *Circulation: Cardiovascular Quality and Outcomes*, vol. 14, no. 8. 2021. doi: 10.1161/CIRCOUTCOMES.121.007858.

[59] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records," *Sci Rep*, vol. 6, 2016, doi: 10.1038/srep26094.
[60] R. E. Wright, "Logistic Regression," in *Reading and Understanding Multivariate Statistics*, L. G. Grimm and P. R. Yarnold, Eds. Washington DC: American Psychological Association, 1995, pp. 217–244.
[61] X. Ma *et al.*, "Development and Validation of Prognosis Model of Mortality Risk in Patients with COVID-19," *Epidemiol Infect*, 2020, doi: 10.1017/S0950268820001727.
[62] J. Fürnkranz, "Decision Tree," in *Encyclopedia of Machine Learning*, G. I. Sammut Claude and Webb, Ed. Boston, MA: Springer US, 2010, pp. 263–267. doi: 10.1007/978-0-387-30164-8_204.
[63] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[64] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Mar. 2016, doi: 10.1145/2939672.2939785.
[65] "LightGBM Website," *Microsoft Corporation*, 2022. https://lightgbm.readthedocs.io/en/v3.3.2/ (accessed Oct. 06, 2022).
[66] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.09516
[67] M. Pishgar, M. Razo, J. Theis, and H. Darabi, "Process Mining Model to Predict Mortality in Paralytic Ileus Patients," Aug. 2021, [Online]. Available: http://arxiv.org/abs/2108.01267
[68] E. P. Weledji, "Perspectives on paralytic ileus," *Acute Medicine & Surgery*, vol. 7, no. 1, Jan. 2020, doi: 10.1002/ams2.573.
[69] J. Leonard, "Ileus: Symptoms, causes, treatment, and recovery," Y. (Joe) Soliman MD, Ed. 2021. Accessed: Oct. 06, 2022. [Online]. Available: https://www.medicalnewstoday.com/articles/322149
[70] W. Lu *et al.*, "Causes and prognosis of chronic intestinal pseudo-obstruction in 48 subjects: A 10-year retrospective case series," *Medicine (United States)*, vol. 97, no. 36, 2018, doi: 10.1097/MD.0000000000012150.
[71] F. S. Ahmad *et al.*, "A hybrid machine learning framework to predict mortality in paralytic ileus patients using electronic health records (EHRs)," *J Ambient Intell Humaniz Comput*, vol. 12, no. 3, pp. 3283–3293, Mar. 2021, doi: 10.1007/s12652-020-02456-3.
[72] Team Keras, "Keras documentation: RMSPROP," 2012. https://keras.io/api/optimizers/rmsprop/ (accessed Oct. 04, 2022).
[73] M. F. Sani, S. J. van Zelst, and W. van der Aalst, "Improving the performance of process discovery algorithms by instance selection," *Computer Science and Information Systems*, vol. 17, no. 3, pp. 927–958, Oct. 2020, doi: 10.2298/CSIS200127028S.
[74] J. C. Brown, T. E. Gerhardt, and E. Kwon, *Risk Factors For Coronary Artery Disease*. 2022. Accessed: Oct. 06, 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32119297/
[75] A. E. W. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Sci Data*, vol. 3, May 2016, doi: 10.1038/sdata.2016.35.

[76] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, 2014.
[77] M. Pishgar, S. F. Issa, M. Sietsema, P. Pratap, and H. Darabi, “Redeca: A novel framework to review artificial intelligence and its applications in occupational safety and health,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 13. MDPI, Jul. 01, 2021. doi: 10.3390/ijerph18136705.
[78] R. Perrault *et al.*, “The AI Index 2019 Annual Report,” *Human-Centered AI Institute, Stanford University*, 2019.
[79] J. Howard, “Artificial intelligence: Implications for the future of work,” *American Journal of Industrial Medicine*, vol. 62, no. 11. 2019. doi: 10.1002/ajim.23037.
[80] A. M. Turing, “Lecture to the London Mathematical Society on 20 February 1947. 1986.,” *MD Comput*, vol. 12, no. 5, 1995.
[81] S. L. Andresen, “John McCarthy: Father of Al,” *IEEE Intelligent Systems*, vol. 17, no. 5. 2002. doi: 10.1109/MIS.2002.1039837.
[82] S. A. Akbar, S. Chattopadhyay, N. M. Elfiky, and A. Kak, “A Novel Benchmark RGBD Dataset for Dormant Apple Trees and Its Application to Automatic Pruning,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2016. doi: 10.1109/CVPRW.2016.50.
[83] J. Ding, J. Wang, N. Yuan, and Q. Pan, “The monitoring system of leakage accidents in crude oil pipeline based on ZigBee technology,” in *2011 2nd International Conference on Mechanic Automation and Control Engineering, MACE 2011 - Proceedings*, 2011. doi: 10.1109/MACE.2011.5987303.
[84] S. E. Delabrida, T. Dangelo, R. A. R. Oliveira, and A. A. F. Loureiro, “Towards a wearable device for monitoring ecological environments,” in *Proceedings - 2015 Brazilian Symposium on Computing Systems Engineering, SBESC 2015*, 2016. doi: 10.1109/SBESC.2015.35.
[85] A. Shukla and H. Karki, “Application of robotics in offshore oil and gas industry-A review Part II,” *Rob Auton Syst*, vol. 75, 2016, doi: 10.1016/j.robot.2015.09.013.
[86] G. Belforte, P. Gay, and D. Ricauda, “Robotics for Improving Quality, Safety and Productivity in Intensive Agriculture: Challenges and Opportunities,” in *Industrial Robotics: Programming, Simulation and Applications*, 2006. doi: 10.5772/4923.
[87] A. L. Buczak, H. Wang, H. Darabi, and M. A. Jafari, “Genetic algorithm convergence study for sensor network optimization,” *Inf Sci (N Y)*, vol. 133, no. 3–4, 2001, doi: 10.1016/S0020-0255(01)00089-5.
[88] H. Liu, H. Darabi, and P. Banerjee, “A New Rapid Sensor Deployment Approach for First Responders,” 2005.
[89] M. Pishgar, F. Karim, S. Majumdar, and H. Darabi, “Pathological Voice Classification Using Mel-Cepstrum Vectors and Support Vector Machine,” Dec. 2018, [Online]. Available: http://arxiv.org/abs/1812.07729
[90] F. Barboza, H. Kimura, and E. Altman, “Machine learning models and bankruptcy prediction,” *Expert Syst Appl*, vol. 83, 2017, doi: 10.1016/j.eswa.2017.04.006.
[91] M. Mann and M. Smith, “Automated Facial Recognition Technology: Recent Developments and Approaches to Oversight,” *University of New South Wales Law Journal*, vol. 40, no. 1, 2017, doi: 10.53637/kavv4291.

[92] T. Davenport, A. Guha, D. Grewal, and T. Bressgott, “How artificial intelligence will change the future of marketing,” *J Acad Mark Sci*, vol. 48, no. 1, 2020, doi: 10.1007/s11747-019-00696-0.
[93] N. Kumar, R. Gupta, and S. Gupta, “Whole Slide Imaging (WSI) in Pathology: Current Perspectives and Future Directions,” *Journal of Digital Imaging*, vol. 33, no. 4. 2020. doi: 10.1007/s10278-020-00351-z.
[94] D. D. Rhoads, “Computer vision and artificial intelligence are emerging diagnostic tools for the clinical microbiologist,” *Journal of Clinical Microbiology*, vol. 58, no. 6. 2020. doi: 10.1128/JCM.00511-20.
[95] J. Reason, “The contribution of latent human failures to the breakdown of complex systems.,” *Philos Trans R Soc Lond B Biol Sci*, vol. 327, no. 1241, 1990, doi: 10.1098/rstb.1990.0090.
[96] U.S. Bureau of Labor Statistics, “Census of Fatal Occupational Injuries (CFOI),” 2020. https://www.bls.gov/iif/oshcfoi1.htm (accessed Oct. 05, 2022).
[97] K. Schwab, “The_Fourth_Industrial_Revolution,” 2017.
[98] S. L. Tamers *et al.*, “Envisioning the future of work to safeguard the safety, health, and well-being of the workforce: A perspective from the CDC’s National Institute for Occupational Safety and Health,” *American Journal of Industrial Medicine*, vol. 63, no. 12. 2020. doi: 10.1002/ajim.23183.
[99] Centers for Disease Control and Prevention (CDC): The National Institute for Occupational Safety and Health (NIOSH), “Future of Work Initiative,” 2020. https://www.cdc.gov/niosh/topics/future-of-work/default.html (accessed Oct. 05, 2022).
[100] European Commission, “Shaping Europe’s digital future: Commission presents strategies for data and Artificial Intelligence,” *Https://Ec.Europa.Eu/Commission/Presscorner/Detail/En/Ip_20_273*, no. February, 2020.
[101] U.S. Bureau of Labor Statistics, “Number and rate of fatal work injuries, by industry sector,” *Graphics for Economic News Releases*, 2020. https://www.bls.gov/charts/census-of-fatal-occupational-injuries/number-and-rate-of-fatal-work-injuries-by-industry.htm (accessed Oct. 06, 2022).
[102] S. F. Issa, K. Patrick, S. Thomson, and B. Rein, “Estimating the number of agricultural fatal injuries prevented by agricultural engineering developments in the United States,” *Safety*, vol. 5, no. 4, 2019, doi: 10.3390/safety5040063.
[103] J. Gomez-Gil, I. San-Jose-Gonzalez, L. Fernando Nicolas-Alonso, and S. Alonso-Garcia, “Steering a Tractor by Means of an EMG-Based Human-Machine Interface,” *Sensors*, vol. 11, pp. 7110–7126, 2011, doi: 10.3390/s110707110.
[104] J. Szczepaniak, W. Tanaś, T. Pawłowski, and J. Kromulski, “Modelling of agricultural combination driver behaviour from the aspect of safety of movement,” *Annals of Agricultural and Environmental Medicine*, vol. 21, no. 2, pp. 403–406, 2014, doi: 10.5604/1232-1966.1108613.
[105] M. Zanatta, F. G. Amaral, and G. Vidor, “The role of whole-body vibration in back pain: A cross-sectional study with agricultural pilots,” *Int J Ind Ergon*, vol. 74, p. 102872, Nov. 2019, doi: 10.1016/J.ERGON.2019.102872.
[106] A. M. Kociolek, A. E. Lang, C. M. Trask, R. M. Vasiljev, and S. Milosavljevic, “Exploring head and neck vibration exposure from quad bike use in agriculture,” *Int J Ind Ergon*, vol. 66, pp. 63–69, Jul. 2018, doi: 10.1016/J.ERGON.2018.02.009.

[107] A. Calvo, E. Romano, C. Preti, G. Schillaci, and R. Deboli, “Upper limb disorders and hand-arm vibration risks with hand-held olive beaters,” *Int J Ind Ergon*, vol. 65, pp. 36–45, May 2018, doi: 10.1016/J.ERGON.2018.01.018.
[108] M. Pishgar, G. R. Aby, S. F. Issa, and H. Darabi, “Application of REDECA Framework to Improve the Safety and Health of the Agricultural Tractor Drivers,” 2023.

APPENDIX

Permissions to reprint articles in this dissertation have been granted by the following pubishers:

REDECA: A Novel Framework to Review Artificial Intelligence and Its Applications in Occupational Safety and Health was published by the International Journal of Environmental Research and Public Health under MDPI OpenAccess. The MDPI Open Access Information and Policy states:

> "No special permission is required to reuse all or part of article published by MDPI, including figures and tables. For articles published under an open access Creative Common CC BY license, **any part of the article may be reused without permission provided that the original article is clearly cited**."
> (https://www.mdpi.com/openaccess, accessed January 17, 2023)

Prediction of Unplanned 30-day Readmission for ICU Patients with Heart Failure and *Process Mining- Deep Learning Approach to Predict Survival Outcome for COVID-19 Patients* were published by the BMC Medical Informatics and Decision Making under open access. BMC "Reprints and permissions" policy notes:

> Open access articles
>
> The **open access articles published in BMC's journals** are made available under the Creative Commons Attribution (CC-BY) license, which means they are accessible online without any restrictions and **can be re-used in any way, subject only to proper attribution** (which, in an academic context, usually means citation)."

(https://www.biomedcentral.com/about/policies/reprints-and-permissions, accessed January 17, 2023)

Process Mining Model to Predict Mortality in Paralytic Ileus Patients was published in the proceedings of the 2021 International Conference on Cyber-Physical Social Intelligence (ICCSI) under the IEEE. *Improving Process Discovery Algorithms Using Event Concatenation* was published by IEEE Access. The IEEE reuse policy is as follows:

> "Can I reuse my published article in my thesis?

<u>You may reuse your published article in your thesis or dissertation without requesting permission</u>, provided that you fulfill the following requirements depending on which aspects of the article you wish to reuse.

Text excerpts: Provide the full citation of the original published article followed by the IEEE copyright line: © 20XX IEEE. If you are reusing a substantial portion of your article and you are not the senior author, obtain the senior author's approval before reusing the text.
Graphics and tables: The IEEE copyright line (© 20XX IEEE) should appear with each reprinted graphic and table.
Full text article: Include the following copyright notice in the references: "© 20XX IEEE. Reprinted, with permission, from [full citation of original published article]."
…
Only the accepted version of your article, not the final published version, may be posted online in your thesis."

(https://magazines.ieeeauthorcenter.ieee.org/choose-a-publishing-agreement/avoid-infringement-upon-ieee-copyright/, accessed January 17, 2023)

VITA

NAME	Maryam Pishgar
EDUCATION	M. Eng., Industrial Engineering, University of Illinois at Chicago, Chicago, 2021
	B. Eng., Industrial Engineering, University of Mazandaran, Iran, 2012
EMPLOYMENT	Adjunct Professor, Computer Engineering and Computer Science, Cal State Long Beach, Fall 2022 – present
	Adjunct Professor, Industrial Engineering and Computer Information Systems, Cal Poly Pomona, Spring 2022 – present
	Ph.D. Student, Process Mining and Intelligent Systems Analytics Team (PROMINENT), University of Illinois at Chicago, Chicago, IL, United States, 2017 - present
	Statistics Tutor, Harold Washington College, Chicago, 2012- 2021
PUBLICATIONS	
Journal	M. Pishgar, S.F. Issa, M. Sietsema, P. Pratap, H. Darabi, REDECA: A Novel Framework to Review Artificial Intelligence and Its Applications in Occupational Safety and Health. Int. J. Environ. Res. Public Health 2021, 18, 6705
	M. Pishgar, J. Theis, M. Del Rios, A. Ardati, H. Anahideh, H. Darabi. Prediction of Unplanned 30-day Readmission for ICU Patients with Heart Failure, BMC medical informatic and decision making, May 2, 2022
	M. Pishgar, S. Harford, J. Theis, W. Galanter, JM. Rodríguez-Fernández, K. Chow K, KM. Kochendorfer, Zulueta J, H. Darabi. Process Mining- Deep Learning Approach to Predict Survival Outcome for COVID-19 Patients, BMC medical informatic and decision making, July 25, 2022
	M. Pishgar, M. Razo, H. Darabi, Improving Process Discovery Algorithms Using Event Concatenation, IEEE Access, June 22, 2022

W. Galanter, J. M. Rodríguez-Fernández, K. Chow, S. Harford, K. M. Kochendorfer, M. Pishgar, ... & H. Darabi, Predicting clinical outcomes among hospitalized COVID-19 patients using both local and published models. BMC Medical Informatics and Decision Making 2021, 21(1), 1-18

Conference

M. Pishgar, F. Karim, S. Majumdar, H. Darabi.
Pathological voice classification using mel-cepstrum vectors and support vector machine, IEEE 2018 big data.

M. Pishgar, M. Razo, J. Theis, H. Darabi.
Process Mining Model to Predict Mortality in Paralytic Ileus patients, International Conference on Cyber-physical Social Intelligence 2021.

Pishgar, M., Darabi, H.
Process Mining/ Deep Learning Model to Predict Mortality in Coronary Artery Disease Patients, International Conference on Cyber-Physical Social Intelligence 2022.

In Preparation

Pishgar, M., Issa, S.F.; Aby, R.G..; Darabi, H., Application of REDECA Framework to Improve the Safety and Health of the Agricultural Tractor Drivers, Int. J. Environ. Res. Public Health (Under review).

Pishgar, M., Darabi, H., Effect of a Process Mining based Pre-processing Step in Prediction of the Critical Health Outcomes, IEEE Biomedical and Health Informatics (Under review).