UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Prognostic prediction models using Self-Attention for ICU patients developing acute kidney injury

Pedro Miguel Pereira Domingues

Mestrado em Ciência de Dados

Dissertação orientada por:

Professor Nuno Ricardo da Cruz Garcia

2022

"If you torture the data long enough, it will confess"

— Ronald H. Coase, *Essays on Economics and Economists*

Acknowledgements

To start I must thank my supervisors Prof. Sara C. Madeira for helping in what she could, Prof. Sofia Teixeira for helping me during Prof. Sara's absence, and specially Prof. Nuno Garcia, that accepted the task of guiding the last steps of development of this thesis.

Thank you to my family for constantly making me want to push myself forward, always with the meaningful goal of making you proud, sometimes unconsciously.

A word of appreciation to all my colleagues from Estatística Aplicada, that made this academic experience the best years of my life. A special thanks to Chora, Frota and Satélite for everything we've shared throughout our way to (almost) adulthood. Honorable mentions to Alex, Mourinho and Zlatan as well.

A big thanks to all my friends from OVL, that also matured with me despite our different academic and professional paths. During these years, we've lived some incredible memories that will always be referenced when we're all together.

Last but not least, I want to thank Margarida. Four years and counting, we've been side by side through tough times including a global pandemic. You've been my anchor and I want to thank you for your patience during this time. Thank you for everything, you've made everything easier.

"Acknowledging the good that you already have in your life is the foundation for all abundance." (Eckhart Tolle). Every person mentioned in this section has its own level of importance in my life. Each one had the power of influencing me in what I am today, and while I'll continue to grow, I'm always going to be thankful for having you in my life.

Resumo Alargado

O crescimento geral dos dados e a melhoria da acessibilidade relativa aos registos de saúde eletrónicos (RSE) exigem um nível idêntico de progresso da comunidade de investigação em relação aos modelos clínicos. O uso de técnicas de aprendizagem automática é fundamental para este desenvolvimento, e por isso estão a ser cada vez mais utilizadas em grandes bases de dados médicas com o objetivo de criar soluções que funcionem para pacientes específicos, independentemente da tarefa ou da doença.

A insuficiência renal aguda (IRA) é uma doença definida por mudanças abruptas na função renal, e apresenta alta morbidade e mortalidade, com especial incidência em pacientes em estado crítico. O risco de aparecimento da doença é maior no caso de pacientes com idades mais avançadas (65 anos ou mais), especialmente com um historial anterior de complicações renais, ou igualmente para pacientes expostos a fatores predisponentes tais como sepsis ou grandes cirurgias. Para além das graves implicações a curto prazo, a ocorrência de IRA está associada também a mortalidade a longo prazo, e mesmo os pacientes que sofreram da doença com menor gravidade estão associados a uma maior mortalidade, dado que a reincidência da doença pode acontecer. Nos casos dos pacientes que sobreviveram à ocorrência de IRA, estudos indicam que 41.2% não conseguiram recuperar as funções renais na totalidade, e cerca de 60% desses pacientes acabaram por morrer, um número três vezes superior comparando com os casos de pacientes que recuperaram na totalidade as suas funções renais. Outras repercussões, tais como o aumento do risco cardiovascular, estão também associadas a IRA.

Estas consequências graves demonstram a necessidade de agir rápido por parte dos profissionais de saúde, e isso pode acontecer se existir uma previsão acertada do agravamento da doença no paciente. Uma rápida tomada de decisão pode ser uma questão de vida ou de morte, sendo que a melhor solução é a iniciação da terapia de substituição da função renal (TSFR).

Dada a sensibilidade da doença, nos primeiros estudos realizados houve uma disparidade muito grande relativamente aos resultados obtidos, onde tanto a nível de previsão da ocorrência da doença como a nível da mortalidade se sentiam variações conforme as definições escolhidas para a identificação do estágio. Vários sistemas de classificação diferentes foram sendo utilizados ao longo do tempo, até a escolha se ter fixado no sistema de classificação KDIGO (Kidney Disease: Improving Global Outcome) para a maioria dos estudos mais recentes. Este sistema de classificação, bem como os outros anteriormente utilizados, foca-se nos valores de creatinina (SCr) e de produção de urina (UO) para determinar o estágio da doença no paciente.

Dados da base de dados MIMIC-III foram usados para recolher informações sobre os pacientes. MIMIC-III contém informação referente a pacientes admitidos no Beth Israel Deaconess Medical Center (BIDMC) em Boston, e é uma base de dados pública cuja utilização está apenas sujeita ao preenchimento de um requerimento. Tendo em conta o potencial que dados hospitalares possuem, especialmente referente à capacidade de acompanhar o paciente e de prever possíveis agravamentos ao ponto dos profissionais de saúde poderem atuar a tempo, MIMIC-III é vista como capaz de elevar a comunidade de investigação. A falta de reprodutibilidade nos estudos relacionados com a saúde sempre foi uma crítica feita

pela comunidade científica, e para além do facto de ser de acesso gratuito, a grande variedade de informação sobre os pacientes nesta base de dados abre as portas a diversos tipos de estudos, possibilitando um progresso na investigação científica.

Com o propósito de selecionar pacientes elegíveis para o estudo em questão, foram aplicados critérios de exclusão detalhados. Esses critérios envolveram a idade dos pacientes, tempo da estadia nos cuidados intensivos e principalmente medições tanto de creatinina (SCr) como de produção de urina (UO). Após a exclusão de pacientes, foi igualmente feito um longo processo de exclusão para as variáveis presentes nos pacientes. *Missing Data imputation* foi aplicado nos dados de maneira a ter informação de hora a hora para cada variável, possibilitando a extração de diversas sequências de treino através da sequência completa da estadia do paciente, com o número de horas de acordo com a sua estadia na UCI. Foram extraídas sequências de 6h, 12h e 24h de duração.

As previsões neste trabalho foram feitas utilizando duas variações do sistema de classificação KDIGO: uma onde apenas os valores de SCr foram considerados para determinar o estágio de IRA do paciente (denominado como sistema de classificação sCr para facilitar a escrita), e outra onde tanto SCr como UO foram utilizados (denominado 2B). Embora a maioria dos estudos que abordam IRA apenas usarem os valores de SCr para determinar a condição da doença dos pacientes, os resultados obtidos por ambas as aproximações foram comparados. Esses pacientes foram avaliados em termos de estágios de IRA, com o objetivo de prever o próximo valor do estágio da doença uma hora após a sequência de informação alimentadas ao modelo. Para além de prever o estágio exato da doença, é também analisada a capacidade do modelo não só em prever o agravamento da doença, como também a capacidade de prever a ocorrência de IRA em pacientes sem a doença diagnosticada na hora da previsão.

Mecanismos de *self-attention* foram utilizados para fazer as previsões, através de uma adaptação para séries temporais multivariadas construída a partir de modelos usados com sucesso em tarefas de processamento de linguagem natural (PNL). Para além dos bons resultados em tarefas de PNL, o modelo de *self-attention* usado neste estudo produziu bons resultados em estudos clínicos, ultrapassando assim os resultados obtidos por redes neurais recorrentes (RNNs) em ambas as situações. Comparativamente com RNNs, a quantidade de computação que é feita em paralelo utilizando o modelo de *self-attention* permite um treino muito mais rápido, juntando ao facto de obter melhores resultados.

Nas experiências finais, para todas as experiências exibidas, foi utilizada uma arquitectura diferente da original devido a uma melhoria nos resultados. Para além de haver a comparação de resultados conforme o tamanho de sequências usado, outros pormenores foram testados, com maior foco para o número de variáveis. Para cada um dos sistemas de classificação utilizados, sCr e 2B, as 10 *features* mais importantes foram identificadas através do uso de *Feature Importance*, e foram testados os resultados com o modelo a usar todas comparativamente com apenas as 10 melhores. Os resultados obtidos para 2B foram melhores, obtendo 68.05% de eficácia no que toca à previsão de um episódio de IRA, comparado com os 66.67% de eficácia obtidos em sCr. Para ambos os casos, os resultados foram superiores ao estado da arte.

Este trabalho teve como propósito estudar o desenvolvimento da lesão renal aguda na hora seguinte e compreender a capacidade de modelos de *self-attention* em fazer essas previsões com a eficácia necessária dada a sua importância no contexto do problema, tendo em conta o seu potencial no que toca a problemas relacionados com saúde. A utilização de modelos de *self-attention* bem como as previsões feitas com a definição do estágio de IRA utilizando medidas de SCr e UO podem vir a ter um grande impacto no futuro em estudos de controlo e previsão da doença, sabendo que ataca tão rapidamente.

Palavras Chave: Insuficiência Renal Aguda; Prognóstico; MIMIC-III; *Self-Attention*; *Feature Importance*

Abstract

The general growth and improved accessibility to electronic health records demands an identical level of progress in terms of the research community regarding clinical models. The usage of machine learning techniques is key to this development, and so they are increasingly being used in large medical databases with the purpose of creating solutions that work for specified patients, no matter the task or the disease.

Acute kidney injury (AKI) is a broad disease defined by abrupt changes in renal function. AKI has a high morbidity and mortality, with an increased focus on critically ill patients. The main goal of this thesis is to study the development of AKI within a patient's stay in the intensive care unit (ICU).

Data from the MIMIC-III database was used to collect information regarding the patients. After a detailed exclusion criteria, those were evaluated in terms of AKI stages, with the purpose of predicting the next value of AKI stage one hour after the sequence of information fed to the model. This can suggest the capacity of the model at predicting the aggravation of a patient's AKI condition. The sequences used have hourly information for every feature, and were used sequences of 6h, 12h and 24h length. Self-attention mechanisms were used to make the predictions, using an adaptation for multi-variate time series built from the successfully used models on natural language processing (NLP) tasks.

The predictions on this work were made for two variations of the KDIGO classification system: one where only the serum creatinine (SCr) criteria was taken into account to determine the patient's AKI stage, and other where both SCr and urine output (UO) were considered. While most works addressing AKI only tend to use SCr values to determine the patient's AKI condition, the results were compared using both approaches and were better when using both SCr and UO. For those experiments, the model achieved up to 68.05% accuracy predicting an episode of AKI, compared to the 66.67% accuracy achieved using only SCr values, which outperformed state-of-the-art results for both cases.

Feature importance was also used for each dataset associated with the two variations of KDIGO classification system to identify what were the most important features. Furthermore, final results were compared when using all features versus only using the most 10 important ones.

Keywords: Acute Kidney Injury; Prognostic Prediction; MIMIC-III; Self-Attention; Feature Importance

Contents

List of Figures

List of Tables

Acronyms

2B	Classification System generated through the alterations from 2B Raw
2B Raw	Classification System with the SCr and UO evaluation, following the criteria
AD	Alzheimer's Disease
AKI	Acute Kidney Injury
AKIN	Acute Kidney Injury Network
ALS	Amyotrophic Lateral Sclerosis
ARF	Acute Renal Failure
AUC	Area Under ROC Curve
BIDMC	Beth Israel Deaconess Medical Center
CABG	Coronary Artery Bypass Graft
CK	Creatinine Kinetics
CKD	Chronic Kidney Disease
DOB	Date of Birth
eGFR	Estimated Glomerular Filtration Rate
EHRs	Electronic Health Records
ESRD	End-Stage Renal Disease
FCM	Fuzzy c-means
GFR	Glomerular Filtration Rate
GK	Gustafson-Kessel algorithm
HIPAA	Health Insurance Portability and Accountability Act
ICU	Intensive Care Unit
ID	Identification code for each variable
KDIGO	Kidney Disease: Improving Global Outcome
LOCF	Last Observation Carried Forward
LR	Linear Regression
LSTM	Long-Short Term Memory
MCI	Mild Cognitive Impairment
MDRD	Modification of Diet in Renal Disease
MIMIC-III	Medical Information Mart for Intensive Care
NB	Naive Bayes
NIV	Non-Invasive Ventilation

NLLLoss	Negative Log Likelihood Loss
NLP	Natural Language Processing
NOCB	Next Observation Carried Backward
ReLu	Rectified Linear Unit
RF	Random Forest
RIFLE	Risk, Injury, Failure; Loss, End-Stage Renal Disease
RNN	Recurrent Neural Network
RRT	Renal Replacement Therapy
SAnD	Simply Attend and Diagnose Architecture
Seq2Seq	Sequence-to-Sequence
sCr	Classification System that only uses SCr values for the evaluation
SCr	Serum Creatinine
SFS	Sequential Forward Selection
SI	International System of Units
TS	Takagi-Sugeno Fuzzy Modelling
UO	Urine Output
VANT	Vancomycin-associated Nephrotoxicity

Chapter 1

Introduction

In today's world, we live in the age of data. Everything around us is connected to data sources, and a considerable amount in our lives is digitally recorded. Different data sources provide different types of information, such as structured and unstructured data. Structured data from relational databases, when the information is highly organized and easily accessed, and unstructured data when working with information collected from sensors, which tend to be much more difficult to capture, process and analyze [4]. Some data types can be labelled as business data, cybersecurity data, smartphone data, social media data, health data etc., and with the abundance of available data generated, researchers and analysts are challenged to create and develop procedures capable of augmenting the value of the information at display. Smart city, for example, is a concept where a technologically modern urban area collects specific data through the use of electronic devices, with the ultimate goal of improving operations across the city. Collecting information from citizens, devices or buildings, fed to a well structured integration network [5], would allow an optimized management of several community services, one of them being healthcare [6]. With the continuous increase of population in high density metropolitan areas, the need for health related services to have infrastructures and digital systems capable of dealing with the demand is vital. Thus, getting the most out of the electronic health records (EHRs) at disposal is key, providing real time patient-centered records that make information available instantly and securely to authorized users. Besides containing the complete information, including medical history, diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory and test results for each patient, EHRs also allow access to evidence-based tools that providers can use to make decisions about a patient's care [7]. This shows the importance of data management tools, as they have the capacity of extracting insights and present useful knowledge from data in an intelligent way, potentially making our everyday life easier.

In parallel to the increase of data availability, the computational area is consequently evolving. Machine learning is growing rapidly in regards to data analysis, providing systems with the ability to learn and enhance from experience automatically without any specific programming. To get the best out of real-world applications, machine learning algorithms are used to intelligently analyze the data. It is possible to see machine learning in everyday-life applications such as: e-mail spam filters that separate e-mails based on its importance, fraud detection by banks where they alert the client if some suspicious transaction happened, music streaming apps that suggest specific playlists based on the previous songs and genres the user consumed before, and the several features from social media services, where users have personalized ads, completely customized explore pages (or similar concept, depending on the platform) based on the content the user watched before, and face recognition, either on using filters to take photos, where the face features are instantly recognized, or by the platform automatically identifying people in

the photo based on the user's friend list.

The potential of machine learning is endless, thus it is being used on critical areas such as the health field, where the use of machine learning approaches to target clinical problems is set to revolutionize clinical decision making [8]. The success of these applications depend on the understanding of the intrinsic processes used during the classical pathway by which clinicians make decisions in routine practice. Thus, the real value is on the conjugation of both standard clinical decision making with the machine learning tools. Those tools can have an impact at the levels of: data acquisition, by extracting standardized, high quality information with the less computational cost possible. Feature extraction, by getting the most out of the raw measurements collected by the healthcare practitioners, while also reducing the dimensionality of the data. There's also an impact at the interpretation level, as the machine learning tools can enhance the understanding of patient's clinical status through the handling of complex data. Decision support is where the gold might be, with the ability to predict clinical outcomes and responses to specific treatments, while recommending procedures that could help the clinicians massively on their real-time decision task [8]. The desire of taking advantage of the large amount of medical data that is currently accessible is to build prediction models, using computational intelligence, in order to improve patient care, as these models are able to interpret information, learn rules, or link variables that may not have an obvious correlation.

Accurately predicting the course of a disease opens a window of opportunity for the caretaker to intervene before it progresses. In order to achieve good discrimination capacity to predict patient outcomes, it is important that the models are trained with subsets of patients similar to the patients of interest. Therefore, it is common practice to train several models over several cohorts of patients that share important characteristics, instead of training one single model with a pool of potentially very distinct patients [9].

The information measured on clinical data can be analyzed as time series data. Using multiple time series to predict clinical related tasks is not easy, as multiple measurements are collected with distinct time intervals and can influence the overall accuracy if not addressed [10]. Initially developed for natural language processing (NLP), self-attention is capable of achieving state-of-the-art performance for several clinical prediction tasks, while outperforming other deep neural networks in terms of computation costs [3]. Capable of capturing complicated non-linear dependencies across the multiple time series and their time steps [11], the promising results using self-attention mechanisms make this a reliable option for tackling health related tasks. Also, it will be interesting to follow the constant evolution of self-attention architectures by the research community in the near future. The release of new architectures will certainly push the state-of-the-art performance for several clinical related tasks.

1.1 Context and Motivation

Acute kidney injury (AKI) is a broad disease defined by abrupt changes in renal function. Although being a robust organ, kidneys are capable of enduring several levels of abuse. However, when under severe pressure together with some other clinical issues for patients with adverse prognosis there can be an abrupt decrease in renal function [12]. AKI has a high morbidity, with an increase focus on critically ill patients [13], and its development is not directly related to having kidney problems at admission, showing the importance of tracking the patient to avoid worse prognosis. Those who develop AKI have a higher mortality rate, a higher requirement of renal replacement therapy (RRT) and a general longer hospital stay comparing with patients without AKI [12]. The adverse outcomes of having AKI are still dangerous

even when mild events happen, as there is an increase in a 10 years long mortality rate [14]. Besides that, the hospital costs associated with AKI patients are also a problem, as it is known that in the USA the costs for AKI patients double the costs compared to patients who did not develop AKI, and in the UK 1% of all the health and social care budget is related with AKI and its consequences [15, 14]. Knowing this, being capable of identifying the patients set to have the worse possible outcomes before the condition progresses is a an important task. An earlier detection of the condition would grant a much more careful focus on patients of higher risk, allowing the clinicians to adjust treatments while improving resource division [12].

The vast majority of studies regarding AKI address the mortality aspect, specifically the studies conducted by Cunha et al. [16], Correia et al. [17] and Silva et al. [18]. The first studied mortality prediction on short and long-term for patients that survived AKI, the second studied which physiological variables are most predictive of mortality within ICU patients that develop AKI, and the latter developed a model capable of predicting which patients are at larger risk of death, allowing a more efficient use of hospital resources. More detail on these studies will be provided on the related work section (section 2.1.3).

Instead of studying when or how the patient died, the main goal of this thesis is to accurately predict the aggravation of AKI for patients in the ICU, while also determine if self-attention mechanisms can be a valuable asset to procedures like the one used in this work, not only for AKI but for general clinical tasks. The ultimate goal is to improve patient care by supplementing clinicians on information of what could happen and let them decide of whether they should act differently, knowing what is expectable to happen in the following hours.

1.2 Contributions

- This work uses a self-attention model to predict the stage of AKI in a continuous way, working with sequence length data, with the purpose of exploiting the attention mechanism's capacity to deal with multivariate time series. This thesis is, to the best of our knowledge, one of the first works to use self-attention mechanisms to specifically predict the progression of AKI severity for patients in the ICU. The results achieved in this work outperformed the state-of-the-art results when predicting an episode of AKI for patients in the ICU.

- The self-attention model was tested with the original architecture and with different experiments regarding the several architectural details. The results ended up being better using parameters different from the original architecture and were used across all experiments displayed in this work.

- The predictions were made for two different classification systems: one generally used by the community, where the staging criteria only uses the serum creatinine (SCr) values for the staging (labeled as the sCr classification system), and the other one was generated using both SCr and urine output (UO) values (labeled as the 2B classification system). The procedure used to obtain 2B was, to the best of our knowledge, developed for the first time in this work, and produced better results comparing to sCr. This shows that the usage of UO regarding AKI stage aggravation can be an improvement and should be considered on future works.

1.3 Thesis Outline

Chapter 2 gives an insight into AKI and the attention mechanisms. Regarding AKI, a definition of the disease and a literature review of studies are given, not only from AKI, but also studies on other diseases with interesting approaches that could be somewhat replicated in the context of AKI. There is an overview of the burden of AKI in society and the evolution of the staging systems used throughout. An explanation of how attention mechanisms were introduced in machine learning tasks is addressed. Also, the theoretical definition of concepts like encoder-decoder and self-attention are given associated with examples of architectures that were progressively improving the landscape of attention models for diverse areas. Chapter 3 details the model that is used in this thesis, its architecture, and the alterations from the original model. Chapter 4 describes the data. It explains some specifications chosen and criteria used to extract the data and the processing tasks made to reach the final cohort. The procedure to get the data the way it fits the model is shown, as well as the reasons behind the dataset selection for the different approaches. Chapter 5 details the process of discovering the most important features for the different approaches. It also addresses the problem of having an unbalanced dataset. Chapter 6 summarizes the main results and compares the different approaches. Results using the two variations of the KDIGO classification system, also comparing the results when using different lengths of sequences and the number of features. Chapter 7 concludes this work, showing the goals achieved. The limitations are declared and directions for future work in this area are presented.

Chapter 2

Background

2.1 Acute Kidney Injury

Acute Kidney Injury (AKI) is a common episode within patients in the Intensive Care Unit (ICU). Before being called AKI, it was known as Acute Renal Failure (ARF), and was considered to be a high incidence illness associated with increased risk of death. Because there was no standard definition, the studies on ARF presented quite inconstant results: its incidence would vary between 1% to 25% and the mortality rate from 28% to 90% [19]. This disparity result-wise made the studies difficult to compare due to different definitions used, which altered the number of occurrences. In cases that the criteria for ARF definition was more conservative, the incidence decreased, but the correspondent mortality rate increased [20].

Based on evidence from previous studies, it was published the first consensus definition and staging criteria [19], called RIFLE. As this classification system includes the entire spectrum of acute changes in renal function from mild to severe stages, the term AKI took the spotlight [19, 12, 21]. Since then, some other updated classification systems were created, like the AKIN and KDIGO criteria.

AKI is a syndrome, some say even a group of syndromes [22], defined by an abrupt loss of kidney functionalities, specifically an abrupt decrease in glomerular filtration, meaning a general increase in serum creatinine levels (SCr). This syndrome is also highly associated with morbidity, mortality and high costs, as an occurrence of AKI may lead to the development of chronic kidney disease (CKD) or even end-stage renal disease (ESRD), and because of the increasing incidence of AKI, its impact on long-term health and costs is greater than it was expected [23, 24]. That increasing incidence of AKI is similar across countries, with no significant difference between high-income and the low-to-middle-income countries, although the renal replacement therapy (RRT) ends up more widely used on the high-income ones [25].

The risk of having AKI is higher on ICU patients due to their already fragile state. Besides that, the risk is also higher for patients of higher age (65 and above) with a previous record of kidney diseases, or patients that have been exposed to predisposing factors, such as sepsis and major surgeries [25]. Early implementation of RRT is an important asset, as it significantly reduced mortality and enhanced renal recovery at 1 year on patients with KDIGO stage 2 and 3 [26].

Another important topic to address is the fact that an episode of AKI is definitely not only associated with short-term outcomes, as it is also proven to have an impact on long-term survival. Even the patients in the lower stages of the disease are associated with a reduction in survival, remaining detectable for 10 years or more [27]. A study analysed recovery patterns for patients after AKI, where it showed that

41.2% of those patients did not fully recover their renal function before hospital discharge. Out of those, 1-year age-adjusted mortality was nearly 60%, which is more than three times compared to patients who did recover fully. Furthermore, 1-year mortality was lowest (10%) in the group of patients (26.6% of the cohort) whose AKI reversed within 7 days and remained stable until discharge [28]. AKI survivors have an increased risk of developing CKD or ESRD, and even though it hasn't before, it is now widely accepted, but still not well appreciated for other specialties, besides nephrology [29]. Increasing risks of gastrointestinal bleeding, organ fibrosis are also diseases associated with AKI [30, 31] as well as an increased cardiovascular risk observed in patients after AKI [32, 33, 34]. So it is possible to see the repercussions of AKI not only in the present, but also in the near and distant future.

2.1.1 AKI Staging Systems

Each individual patient in the ICU has its own level of severity in terms of AKI, so, detecting AKI in a patient implies the usage of classification methods capable of doing so in the most accurate way possible, using the (sometimes limited) information it disposes. The three main criteria used are RIFLE, AKIN and KDIGO, which base their predictions on levels of serum creatinine and urine output. This can be used to place them within other patients with close characteristics, easing future medical procedures, as it is possible to act similarly for patients on the same stage of AKI. Although others (like CK) have also been used before, with interesting results regarding mortality [35], the usual main options are the three mentioned before.

Urine output values tend to decrease with the severity of the disease stage. The opposite happens with SCr values, which is seen as an insensitive marker of kidney dysfunction, particularly in patients without underlying CKD, because a normal kidney has considerable excess filtration capacity. Even health kidney donors might show very small changes in SCr concentration, meaning that losing large renal mass does not necessarily increase SCr levels.

It is reported the difficulties of predicting the exact stage of the critically ill patients, which as expected, turns out to be more difficult than simply predicting the occurrence of AKI [36]. Depending on the choice for the classification system, the results will come out different due to some patients being selected into distinct groups, therefore creating different cohorts, making it hard and less viable to specifically compare results when applying different classification systems.

Based on Ulger et al.'s work [35], where several AKI classification systems were compared, a summary of the SCr criteria for the different classification systems is shown in Table 2.1, making it clear and simple to see the differences between them, contrarily to the UO criteria, since it's pretty much equal across all of them.

2.1.1.1 RIFLE Classification

In 2004, RIFLE was introduced to the AKI work line, being the first classification system to get consensus within the community [19]. RIFLE is an acronym for the 3 AKI stages defined by this criteria: Risk, Injury and Failure, and also 2 other outcomes: Loss and End-stage Kidney Disease.

This criteria consists on the reduction of glomerular filtration rate (GFR) according to baseline value, which is associated with an increase of serum creatinine levels, and also urine output.

Classification System	Stage	Serum Creatinine criteria	Urine Output criteria
RIFLE	Risk	To ≥1.5 times baseline	<0.5ml/kg/hr for >6 hr
	Injury	To ≥2 times baseline	<0.5ml/kg/hr for >12 hr
	Failure	To ≥2 times baseline or ≥0.5 mg/dl increase to at least 4.0 mg/dl	<0.3ml/kg/hr for >24 hr or Anuria for >12 hr
AKIN	1	Increase of ≥0.3 mg/dl or to 1.5-1.9 times baseline	<0.5ml/kg/hr for >6 hr
	2	To ≥2-2.9 times baseline	<0.5ml/kg/hr for >12 hr
	3	To ≥3 times baseline or ≥0.5 mg/dl increase to at least 4.0 mg/dl or RRT	<0.3ml/kg/hr for >24 hr or Anuria for >12 hr
KDIGO	1	Increase of ≥0.3 mg/dl within 48hr or to 1.5-1.9 times baseline	<0.5ml/kg/hr for 6-12 hr
	2	To ≥2-2.9 times baseline	<0.5ml/kg/hr for ≥12 hr
	3	To ≥3 times baseline or at least 4.0 mg/dl or RRT	<0.3ml/kg/hr for ≥24 hr or Anuria for ≥12 hr

Table 2.1: Diagnosis criteria of AKI according to each classification system

2.1.1.2 AKIN Classification

The Acute Kidney Injury Network proposed in 2007 a revised classification criteria, hence the name, that was based on the RIFLE criteria with particular belief that smaller SCr increases are of prognostic value and takes into account the increase in mortality in the R stage of RIFLE.

Although also being based in SCr and urine output, and divided in 3 stages, their nomenclature was altered to 1, 2 and 3 with increased severity. In AKIN, the GFR measure is no longer relevant, mostly due to GFR not being easily measured and usually estimated using SCr values. So, the definition given only by SCr levels is enough, also making the AKI classification more homogeneous. Another alteration was the time window in which changes had to occur. It was reduced from 7 days to 48 hours during a patient hospital stay.

2.1.1.3 KDIGO Classification

In 2012, the Kidney Disease : Improving Global Outcome published a new classification system, merging both RIFLE and AKIN systems, as an effort to standardize the approaches for AKI diagnostic and treatment. Again, it uses SCr and urine output although with changes in the baseline.

Nowadays, KDIGO tends to be the classification system more used due to its higher capability of accurate prediction of AKI incidence.

2.1.2 Baseline Estimations

Working with AKI implies constantly determining the disease severity for every patient, and as we saw before, all of the three main classification systems use SCr and UO in their criteria. The usage of SCr ends up being more relevant, due to the lack of information regarding the urine output on ICU patients. Since the baseline value is usually missing, there is a need to estimate it, so, defining the baseline for SCr may turn out important because some patients can shift their stage of AKI depending on the baseline chosen. AKI incidence is correlated with the baseline value, which means that patients with higher levels of baseline have more risk of developing AKI, where a poor estimation may lead to a delayed diagnostic [37].

There are several methods for baseline SCr estimation, such as baseline equal to the SCr value at admission [38], equal to the lowest value of SCr [39] or even baseline equal to the lowest value of the

first three measures of SCr[13], and the latter is also used in Silva et al. [18], Correia et al. [17] and Cunha et al.'s [16] studies.

2.1.3 Related Work on AKI

Working with critically ill patients under the AKI work line is not particularly a recent working case. There have been several articles and thesis tackling this issue. Using the same dataset that will be used in this work, MIMIC III [40], a few articles addressed this topic with similar approaches within them. Cunha et al.'s [16] aim was to develop predictive models by exploring the outcomes of AKI in ICU patients and on short and long-term mortality among patients who survived their ICU encounter. In this study, the methods used were Fuzzy modeling, more specifically Takagi-Sugeno (TS) fuzzy models that consist of fuzzy rules where each one of them describes a local input output relation. The approach was based on the Fuzzy c-means (FCM) and Gustafson-Kessel (GK) clustering algorithms to compute the fuzzy partition, with the number of rules, antecedent fuzzy sets and its parameters being determined using fuzzy clustering. A 10-fold cross validation was used, and the results, in general, obtained with the Gustafson-Kessel algorithm outperformed the ones generated using FCM. The fuzzy models implemented are able to predict one-year mortality with an AUC of 0.75 from the moment of the admission, and an AUC of 0.76 with an accuracy of 0.69 for 24-hour ahead prediction.

One year later, Correia et al.'s [17] thesis studied which physiological variables are most predictive of mortality within ICU patients that develop AKI. For that, the authors used sequential forward selection (SFS) and fuzzy modeling to construct a mortality prediction model for each of the 5 cohorts, since the authors divided the data for each stage of AKI, one for all patients with AKI and another for all patients using data from the first and the last day. Just like in the previous study, this one also used TS and FCM, as well as cross validation. The models were also generated with and without sequential forward selection in order to evaluate its influence, and the results of mortality predictions indicated that using SFS had very little impact. Fuzzy models achieved very good results, in general, but the best ones were achieved using information of the last day. For those, the best was for patients with stage 3 with a score of 0.92 ± 0.05 for AUC, 0.83 ± 0.07 accuracy, 0.82 ± 0.08 for sensitivity and 0.84 ± 0.08 for specificity, which does mean a significant improvement compared to Cunha et al.'s article.

The last one is Silva et al. [18], that developed a model capable of predicting which patients are at larger risk of death, allowing a more efficient use of hospital resources. When using the mean values for time series the model achieved AUC values of up to 0.92 in patients with stage 3 AKI, and 0.86 for the general population, showing the potential of using this type of predictive applications for human resources scheduling, as well as management of monetary resources on critical patients according to their need for attention and treatment. Just like the 2 latter studies, cross validation, TS and FCM were used, as the data was also divided into 5 cohorts. The later fuzzy models built were modelled twice, with and without SFS, which did not represent significant alterations in the maximum values obtained since the values were all approximately 0.9. Once again showing that SFS had very little impact on the final results performance, besides being a good option to apply when looking for building models with less variables.

In a brief comparison between the three studies, all of them used both TS and FCM, but only the first one also used GK. The inclusion-exclusion criteria was different between the first study and the latter two, and it is noticeable when looking at the number of features kept for modeling, which were 8, 32 and 28, respectively per chronological order. The different discretization methods used are also relevant to the final results, since the best results obtained in the 2 latter studies were through the use of mean

values for the time series. The first one did not had into account the mean values, as it only focused on the median. This shows the importance of trying different approaches to ensure we get the best results, as it happened in the works of Correia et al. and Silva et al., that tried the mean and last value, besides the median used in Cunha et al.'s study.

In a brief comparison between the three studies, all of them used both TS and FCM, but only the first one also used GK. The inclusion-exclusion criteria was different between the first study and the latter two, and it is noticeable when looking at the number of features kept for modeling, which were 8, 32 and 28, respectively per chronological order, and equally relevant to the final results joint with the discretization method, since the best results obtained in the 2 latter studies were through mean values, the first one did not had into account the mean values, only focusing on the meaning it is important to obtain values for the median, mean and last value to ensure we get the best results and the ability to compare them, which happened in the works of Correia et al. and Silva et al., while in Cunha et al.'s study only the median was used.

Another common aspect between the 3 studies is that all of them used the AKIN classification, which means the most recent classification system, the KDIGO classification system, was excluded or maybe not taken into account. Also, it is noticeable the upgrades and small changes from one study to the next, chronologically wise, availing the good things done before while striving to find better results with their own small changes.

The study by Ulger et al. [35] evaluates the effects of AKI development on mortality in critically ill trauma patients followed in the ICU, from a turkish hospital, with four different classification systems: RIFLE, AKIN, CK and KDIGO. Out of the 198 patients included in the study, when investigated upon an existence of AKI, 74.2% of the patients were identified as having AKI according to the KDIGO criteria, followed by AKIN with 72.2%, RIFLE with 69.7%, and at last CK with 59.1%, that also indicated a significant increase in mortality in patients with AKI diagnosed on the first day. In this study, as expected, the compatibility between RIFLE, AKIN and KDIGO was higher than with CK, but, with an agreement between all four classification systems, the presence of AKI was found to be an independent risk factor in the development of in-hospital mortality.

Sonia Yaqub [41] also compared the three stand-out criteria for their ability to predict all-cause mortality and morbidity after isolated coronary artery bypass graft (CABG) surgery. In general, patients with AKI were older and more likely to be diabetic and hypertensive, and out of the total 1508 Pakistanis: 33.7% were classified as having AKI by the AKIN criteria, 34.4% by the KDIGO criteria and 57.5% by the RIFLE criteria (based on change in estimated GFR, abbreviated to eGFR). AUC for 30 day mortality was 0.786 for AKIN, 0.796 for KDIGO and 0.844 for RIFLE, however, discrimination power for morbidity was low (below 0.7), making it undesirable. This means that AKIN and KDIGO are comparable to estimate AKI, while RIFLE (eGFR based) overestimates its incidence, but has a better discriminatory power in terms of mortality prediction compared to the other two classification systems.

Tomašev et al.[42] developed a deep learning approach for the continuous risk prediction of future AKI in patients. The model was able to predict 55.8% of all inpatient episodes of acute kidney injury, and with a lead time of up to 48h, and a ratio of 2 false alerts for every true alert, the model also predicted correctly 90.2% of all AKI patients that required subsequent administration of dialysis. Besides that, the model provides confidence assessments and a list of clinical features that stand out in each prediction, alongside predicted future trajectories for clinically relevant blood tests. The proposed system is a recurrent neural network (RNN) that sequentially runs over individual electronic health records (EHRs), processing the data one step at a time, building an internal memory that keeps track of the relevant information seen up until that specific point. At each time point, the model outputs a probability of AKI

occurring at any stage of severity within the following 48h, alongside an associated degree of uncertainty for each timestamp in that window. Results wise, sensitivity was high in patients who went on to develop lasting complications following their AKI episode. The model successfully early predicted 84.3% of episodes in which dialysis was required within 30 days of the AKI occurrence (any stage), for in-hospital and outpatients, and also predicted the administration of dialysis scheduled within 90 days for regular outpatients, with 90.2% of success. When only applied to stage 3 AKI, the model correctly predicted 84.1% inpatients up to 48h in advance, higher than the 71.4% when predicting both AKI stages 2 and 3, and also higher than the 55.8% for any AKI stage as seen before, this three results were obtained with a precision of 33% (rate of two false-positives for each true-positive). Analysis of the false-positives alerts indicated that 24.9% were positive predictions that were made even earlier than the 48h window, in patients that later developed AKI. Out of these, 57.1% occurred in patients with pre-existing chronic kidney disease, thus being at higher risk of developing AKI. Of the remaining false-positive alerts, 24.1% were trailing predictions that occurred after the window of prediction. Besides the early and trailing predictions, the model detected that 88% of the patients who did not experience AKI during the admission where the model predicted it to happen were patients with severe renal impairment, known renal pathology or evidence in the EHR that the patient required clinical review. As the authors refer, these alerts can be filtered out during clinical practice.

The data used in their work included medical records registered up to ten years before each patient's ICU admission date, and up to two years after being discharged from the ICU. Whenever available, they were optionally used later as historical features. The final dataset consisted of 707,782 patients, randomly divided into training (80% of observations), validation (5%), calibration (5%) and testing (10%) sets. Each day was broken into four six-hour periods, meaning every patient was represented by a sequence of events, with each one providing information recorded within a six-hour period, grouping together records that have occurred within the same six-hour period. This available data mixed with additional summary statistics and augmentations formed a feature set that was used as input to the predictive models. Each clinical feature was mapped onto a corresponding high-level concept, such as: procedure, diagnosis, prescription, laboratory tests, and many more. In total, 29 high-level concepts were in the data. With the purpose of predicting the risk of developing AKI easily, some features were provided, such as the median yearly creatinine baseline and the minimum 48h creatinine as numerical features, which were used as the baseline values for the KDIGO criteria. Regarding the historical features, 3 historical aggregate feature representations were considered: one for the past 48h, one for the past 6 months and another for the past 5 years. All were optionally provided to the models, and the decision on which combination of historical data to include was based on the model performance on the validation set.

The patient AKI were computed at each time step on the basis of the KDIGO criteria, and out of the three definitions of AKI this classification system accepts (seen on the previous section), only the definitions involving baseline creatinine levels were used to provide ground-truth labels for the onset of AKI, due to the lack of urine output values. Using the KDIGO criteria three AKI categories were obtained: 'all AKI' (KDIGO stages 1, 2 and 3), 'moderate and severe AKI' (KDIGO stages 2 and 3), and 'severe AKI' (KDIGO stage 3). A baseline of median annualized creatinine was used when previous measurements were available, and when these measurements were not present, the modification of diet in renal disease (MDRD) formula was applied to estimate baseline creatinine. The AKI stages were computed every time a serum creatinine measurement was available in the sequence, and then copied forward in time until the next creatinine measurement, where the ground-truth AKI state was updated accordingly. The prediction target at each point in time is a binary variable that is positive if the AKI category of interest (for example, all AKI) occurs within a chosen future time horizon. If no AKI state was

recorded within the chosen horizon, this was interpreted as a negative. A total of 8 future time horizons were used(6-h, 12-h, 18-h, 24-h, 36-h, 48-h, 60-h and 72-h ahead), all available at each time point.

The model itself makes predictions by first transforming the input features using an embedding module. This embedding is fed into a multi-layer recurrent neural network, the output of which at every time point is fed into a prediction module that provides the probability of future AKI at the time horizon for which the model will be trained. To provide useful predictions, an ensemble of predictors are trained to estimate the confidence of the model, and the resulting ensemble predictions are then calibrated using isotonic regression to reflect the frequency of observed outcomes. The embedding layers transform the high-dimensional and sparse input features into a lower-dimensional continuous representation, making subsequent prediction easier. A deep multilayer perceptron was used with residual connections and rectified-linear activation, and L1 regularization on the embedding parameters to prevent overfitting and to ensure that the model focuses on the most-salient features. Recurrent neural networks run sequentially over the electronic health record entries and are able to implicitly model the historical context of a patient by modifying an internal representation (or state) through time. A stacked multiple-layer recurrent network with highway connections between each layer is used, which at each time step takes the embedding vector as an input. The RNN architecture was a simple recurrent unit network, with tanh activations, chosen from a broad range of alternative RNN architectures that did not provide significant improvements, such as: long short-term memory, update gate RNN and intersection RNN, simple recurrent units, gated recurrent units, the neural Turing machine, memory-augmented neural network, the Differentiable Neural Computer and the relational memory core.

After that, the output of the RNN is fed to a final linear prediction layer that makes predictions over all eight future prediction windows (6h windows up until 72h ahead). Each of the resulting eight outputs provides a binary prediction for AKI severity at a specific time window and is compared to the ground-truth label using the cross-entropy loss function (Bernoulli log-likelihood). Besides that, a set of auxiliary numerical predictions were made, where at each step there was a prediction of the maximum future observed value of a set of laboratory tests over the same set of time intervals used to make the future AKI predictions. The laboratory tests predicted are the ones that are known to be relevant to kidney function, such as: creatinine, urea nitrogen, sodium, potassium, chloride, calcium and phosphate. The overall improvement observed for including the auxiliary task was close to 3% area under precision-recall curve in most cases. The overall loss function was the weighted sum of the cross-entropy loss from the AKI predictions and the squared loss for each of the seven laboratory-test predictions. Training-wise, it was used an exponential learning-rate decay, and the best validation results were achieved using backpropagation through time windows. The best performing RNN architecture used a cell size of 200 units per layer and 3 layers, and besides that, an extensive hyperparameter exploration of dropout rates for different kinds of dropout was conducted to determine the best model regularization. Input dropout, output dropout, embedding dropout, cell-state dropout and variational dropout were all tested, but none of them led to improvements, meaning that dropout was not included in the model.

A curated set of clinically relevant features was chosen using existing AKI literature and consensus opinion of six clinicians, along that, 36 of the most-salient features discovered by the deep learning model that were not in the original list were also included, making the final curated dataset contain 315 base features. Also, a set of manually engineered features were computed, as well as a representation of the short-term and long-term history of a patient, resulting in a total of 3,599 possible features for the baseline model.

The best models were evaluated on the independent test set that was retained during model development, and the models selected on the validation set were recalibrated on the calibration set in order to

further improve the quality of the risk predictions. Deep learning models with softmax or sigmoid output trained with cross-entropy loss are prone to miscalibration, so recalibration ensures that consistent probabilistic interpretations of the model predictions can be made. As addressed before, AKI episodes that occur later during in-hospital stay can be predicted earlier than an AKI episode that occurs immediately upon admission, so, to better assess the clinical applicability of the proposed model, the AKI episode sensitivity was computed for different levels of step-wise precision. And since the models were designed for continuous monitoring and risk prediction, the evaluation happened at each 6h time step within all of the admissions for each patient except for the steps within AKI episodes. To gauge uncertainty on the performance of a trained model, 95% confidence intervals were calculated with the pivot bootstrap estimator, by sampling the entire validation and test dataset with replacement 200 times, and to quantify the uncertainty on model predictions (versus overall performance), an ensemble of 100 models was trained with a fixed set of hyperparameters and different initial seeds. The prediction confidence was assessed by inspecting the variance over the 100 model predictions from the ensemble.

Another interesting subject addressed by this study is the subgroup analysis, which is available on the Supplementary Information. This approach is helpful in understanding the performance of predictive models on different clinical sub-populations, as they're not uniform across all population. The different subgroups can be divided in patient demographics, where different age groups, ethnicities and gender are evaluated, in admissions, where the duration of the admission is studied, in patients with chronic kidney disease (CKD), where each CKD stage is evaluated. It also can be divided in a group with other at risk patients, such as diabetic patients and even patients who did not survive after 7 or 30 days upon admission. Besides exploring model performance across the different subgroups, error regression was employed, meaning that for every observation the expected error was computed through logarithmic loss, and fitted a linear regression of the error as an endogenous variable, and population subgroups as exogenous variables. Looking at the results, it is proven that the effect of subgroups on the magnitude of errors is jointly significant.

2.1.4 Related Work on other diseases

Even though each disease has its own progression rate, some of the methods used can also be appropriate to use in the AKI context. Pires et al.[43] worked with Amyotrophic lateral sclerosis (ALS) patients, and in this study it was proposed an approach to stratify the patients in three groups according to their progression rate: Slow, Neutral and Fast, with the purpose of enabling the creation of specialized learning models capable of predicting the need of non-invasive ventilation (NIV), used to prevent respiratory insufficiency as well as proven to improve survival chances on ALS patients. Those models would predict the need of NIV within a time window of 90, 180 and 365 days of their current medical appointment. The models are built using a collection of classifiers and cross validation, while also using FS to test which features are more relevant to the outcome prediction. The progression groups are created from the whole population of patients using information regarding their first symptoms and their first visit. The classifiers are trained for each progression group to predict if a patient will need NIV in k days, and after the models are trained, every time a new patient comes to an appointment the only thing needed is to compute its progression rate in order to identify the corresponding progression group, where the patient's data is used by the specialized model to predict the desired target. The three best classfiers were Naive Bayes (NB), Random Forest (RF) and Linear Regression (LR). Results wise, the number of snapshots (a summary of the patient condition around that time) decreased with the increasing of k, due to

the reducing number of patients. In terms of the feature selection between groups there are differences, with slow progressors needing more features to build good prognostic models, while fast progressors seem to rely on less features. Slow progressors tend to reduce the number of features with increasing time windows, the exact opposite happens with fast progressors, while the neutral seem to maintain the number of features for all time windows. Knowing the selected features in each progression group is important to clinicians, offering the knowledge of which test and exam is the most important for each patient, leading to resource and time optimization, which can lead to a better prognosis. The results achieved up to 0.91 value for AUC for slow progressors in 365 days, and were generally better when using the progression groups compared to the results using with all patients, hence the authors appealing to the usage of patient stratification when studying heterogeneous diseases.

In Pereira et al.'s [44] study, the focus was Alzheimer's disease (AD), known as the most common form of dementia, and also known as non-reversible disease since there is no treatment capable of reverting the progression of the disease [45, 46]. When patients meet criteria for dementia, the brain has already suffered sufficient damage causing severe impact on cognition and autonomy, hence the need to accurately predict beforehand the progression of the disease. Again, like Pires et al.'s study, one of the approaches from this work was the use of supervised learning based on time windows to predict conversion to dementia, labelled as MCI-to-AD conversion (MCI standing for Mild Cognitive impairment), learning from patients stratified on those time windows. The first step of the approach consists in creating the learning examples using time windows, after that, the model and parameters are tuned under a cross-validation scheme, and finally validated using an independent validation set. The model predicts whether a patient diagnosed with MCI at baseline converts to dementia (or if it remains MCI) at time baseline + k (in years), with k corresponding to the considered time window and ranging from 2 to 5.

The innovative strategy used to build learning examples outperformed the common used strategy, named as First Last approach. While the latter used all patients to learn the models, the author's approach grouped patients based on their clinical information, whether they converted (converter MCI) or remained MCI (stable MCI) within a specific time window. The prognostic model used neuropsychological data and was able to predict dementia conversion as early as five years before happening. Besides that first approach, it was also proposed a methodology to address short and long-term (2 and 2-4 years) progression on AD combined with reliability, with the purpose of identifying the trustworthy prognostic models. This second approach consists on a two-step supervised learning, which starts by predicting MCI-to-AD conversion, within a given level of confidence, followed by the prediction of the most likely time window of conversion (once again using short and long-term conversion), using Conformal Prediction. As mentioned by the authors, despite the exploratory results being promising, the small number of examples for long-term converting patients available for training ended up being a setback, thus they pretend to repeat the study whenever more data is available.

Some studies addressed in this section were initially considered as basis for some procedures but ended up not following through, and are explained in the Future Work section (Section 6).

2.2 Attention mechanisms

In psychology, attention is the cognitive and behavioral process of selectively concentrating on one or a few discrete aspects of information while ignoring others. The same way a neural network is considered to be an effort to reproduce and mimic human brain actions in a simplified manner, an attention mechanism

is also an attempt to implement the same action of selectively directing the focus on a few relevant things, paying greater attention to certain factors when processing the data, while ignoring others in deep neural networks [47, 48].

Originally introduced and designed by Bahdanau et al. [49] in 2014, in the context of Neural Machine Translation using Sequence-to-Sequence (Seq2Seq) Models, this mechanism in now used in other tasks besides natural language processing, e.g computer vision [50]. Considered as a natural extension of their previous work on the Encoder-Decoder model [51], this paper laid the foundation of the famous paper ”Attention is All You Need” by Vaswani et al. [2], on transformers that revolutionized the deep learning arena with the concept of parallel processing of words instead of processing them sequentially. This latter publication is at the basis of the article that designed and presented the model that is used in this work, as will be addressed in the section 3.

Still regarding the publication by Bahdanau et al., the core idea is each time the model predicts an output word, it only uses parts of the input where the most relevant information is concentrated instead of the entire sequence, meaning it only pays attention to some input words [52]. This happens as the attention component of the network maps the important and relevant words from the input sentence and assigns higher weights to these words, enhancing the accuracy of the output prediction [47].

The introduction of the attention mechanism improved the performance of the encoder-decoder model for machine translation. The idea behind the attention mechanism was to permit the decoder to utilize the most relevant parts of the input sequence in a flexible manner, by a weighted combination of all the encoded input vectors, with the most relevant vectors being attributed the highest weights. This way, the bottleneck problem that emerges with the use of a fixed-length encoding vector is addressed, where the decoder would have limited access to the information provided by the input. That was specifically problematic for long or complex sequences, as the dimensionality of their representation would be forced to be the same as shorter or simpler sequences [53].

2.2.1 Encoder-Decoder

A Seq2Seq model is a model that receives a sequence of items as input, such as words, letters or time series, and outputs another sequence of items. These models are generally composed of an encoder-decoder architecture, where the encoder processes the input sequence and compresses (encodes) the information into a context vector of fixed length, as said before, and collected in the form of a hidden state vector of any size [1]. This representation is anticipated to be a good summary of the complete input sequence. The decoder is then initialized with this context vector, using which it starts producing the transformed or translated output [52].

In the case of Neural Machine Translation, because the tasks are sequence based, both the encoder and decoder tend to use some form of RNN or Long-Short Term Memory (LSTM). By design, RNNs need two inputs in order to produce an output, meaning that the output at time step t depends on the representation of the previous and current input. The sequential information is preserved in a hidden state of the network and used in the next instance. The encoder, consisting of RNNs, takes the sequence as an input and generates a final embedding at the end of the sequence, that will be used as an input by the decoder. The decoder also uses the previous hidden state to predict the next instance until the end of the sequence, as can be seen in Figure 2.1.

As was addressed before, the size of the sequences is a limitation and that is why attention was introduced. Attention is an interface connecting the encoder and decoder that provides the decoder with

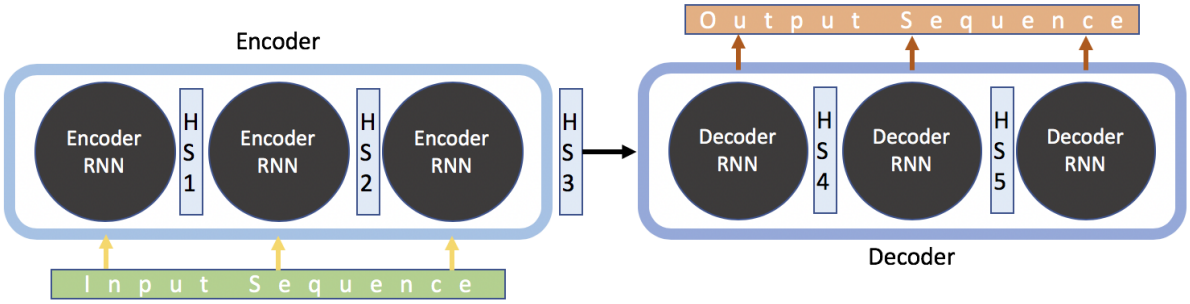


Figure 2.1: Encoder-Decoder model for Seq2Seq modelling (without attention)(extracted from [1])

information from every encoder hidden state, shown in Figure 2.2, instead of just the hidden state from the last encoder instance. Besides that, the other important aspect regarding attention is the context vector, which is generated for every time instance in the output sequences. At every step, the context vector is the weighted sum of the input hidden states (Figure 2.3a).

The generated context vector is combined with the hidden state vector through concatenation (Figure 2.3b), and this new attention hidden vector is used for predicting the output at that time instance. This new attention vector is generated for every time instance in the output sequence, replacing the hidden state vector .

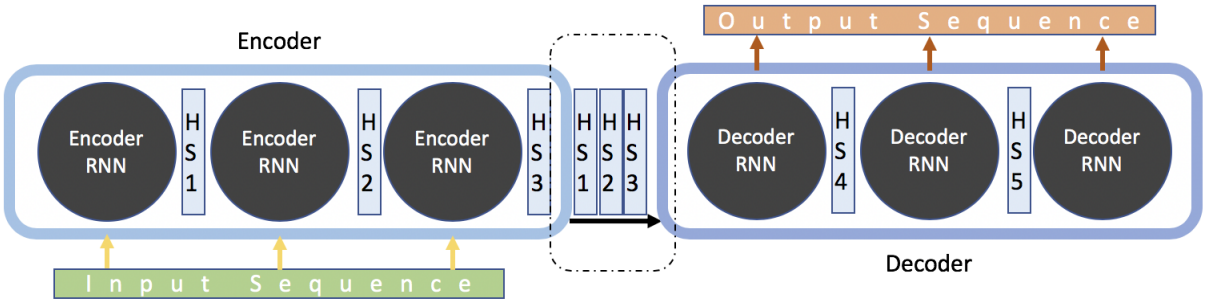
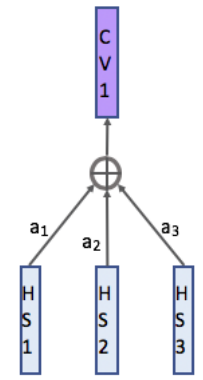
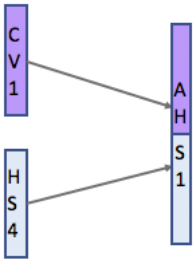


Figure 2.2: Encoder-Decoder model with Attention (extracted from [1])



(a) Context Vector



(b) Attention hidden state

Figure 2.3: Attention component in the Seq2Seq model (extracted from [1])

Regarding the attention scores, these are the output of the alignment model, and score how well an input (represented by its hidden state) matches with the previous output (represented by the attention

hidden state) and does this matching for every input with the previous output. The final step is the usage of a softmax over all these scores, and the resulting values are the attention scores for each input (Figure 2.4. These final values indicate which segment of the input is the most important for each of the instances in the output sequence. The final representation is shown in Figure 2.5.

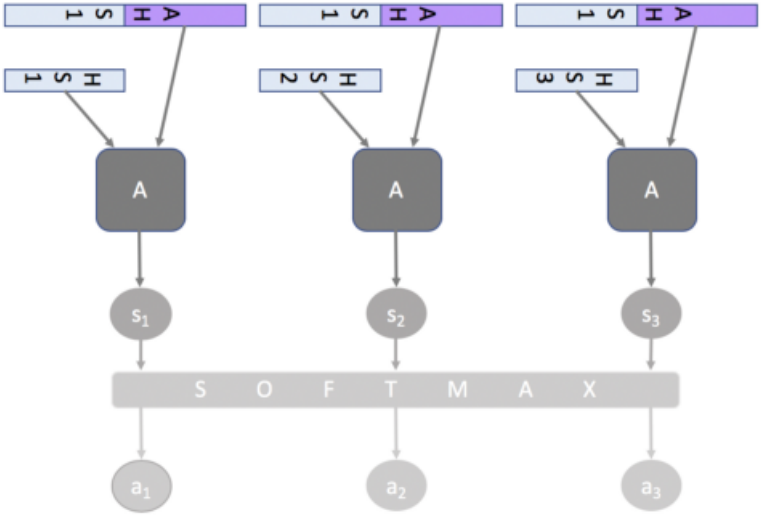


Figure 2.4: Attention scores (extracted from [1])

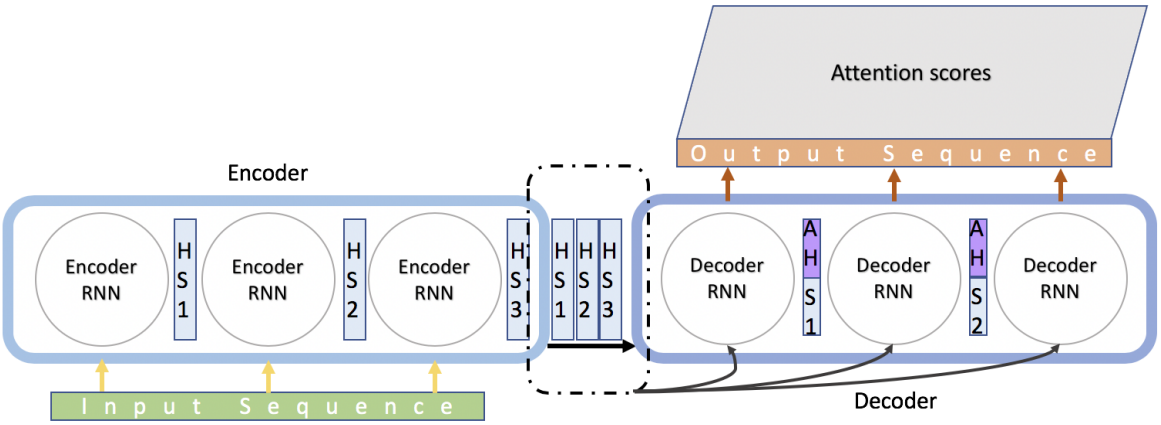


Figure 2.5: Encoder-Decoder model with Attention (extracted from [1])

2.2.2 Transformers

Firstly introduced by Vaswani et al. [2], Transformer-based architectures adopt an encoder-decoder architecture that uses self-attention mechanisms as its main feature to collect global dependencies between inputs and outputs, while completely discarding the use of any recurrence, that was used in former attention mechanisms. Already successfully used in a variety of tasks such as reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [54, 55, 56, 57], self-attention is an attention mechanism that relates to different positions of a single

sequence in order to compute a representation of the full sequence. Comparatively to recurrent and convolutional layers commonly used, the self-attention layers end up needing less total computational complexity per layer, as well as a higher capacity of parallelizable computations, measured by the minimum number of sequential operations required.

Regarding its architecture (Figure 2.6), both encoder and decoder are composed by a stack of N identical layers. In the paper, the authors stack $N = 6$ layers on top of each other for both cases, although nothing is specified about the choice, meaning that using other number would probably be acceptable. Besides being identical in structure, all encoder layers don't share weights, with each layer divided into two sub-layers: a multi-head self-attention mechanism followed by a feed-forward neural network. The encoder inputs go through the self-attention layer, and its outputs are fed to a feed-forward neural network, that is applied independently to each position. The decoder has the same two sub-layers, but between them there is another multi-head attention layer, with a task similar to what attention offers in the Seq2Seq models, meaning it helps the decoder target the relevant parts of the encoder stack output. Also, the self-attention sub-layer is modified in order to prevent it from attending to earlier positions in the output sequence. This is called masking, and on both encoder and decoder sub-layers it's employed residual connections, followed by layer normalization.

As the authors mentioned, "An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query [Q], keys [K], values [V], and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.". This definition makes it easier to understand the Scaled Dot-Product Attention used in the attention layers, whose matrix of outputs is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

The input consists of queries and keys of dimension d_k, and values of dimension d_v. A dot product of the query with all keys is computed, before dividing each by $\sqrt{d_k}$, and applying a softmax function to obtain the weights on the values. In practice, the attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V.

The multi-head attention was implemented as the authors found beneficial to linearly project the queries, keys and values 8 times, and on each of those projected versions the attention function was performed in parallel. This procedure allows the model to attend to information from different representation subspaces at different positions, expanding the model's ability to focus on different positions. It can be represented as:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O, \\ \text{where head}_\text{i} &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.2}$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$, with h being the number of parallel attention layers (heads). Regarding those heads, the dimensions for each one were defined by dividing the full dimensions by h, meaning that the total computational cost is similar to what a single-head attention with full dimensionality would have.

Multi-head attention is used in the Transformer in three different locations in its architecture, as can be seen in Figure 2.6, and the authors explained the purpose of each one as:

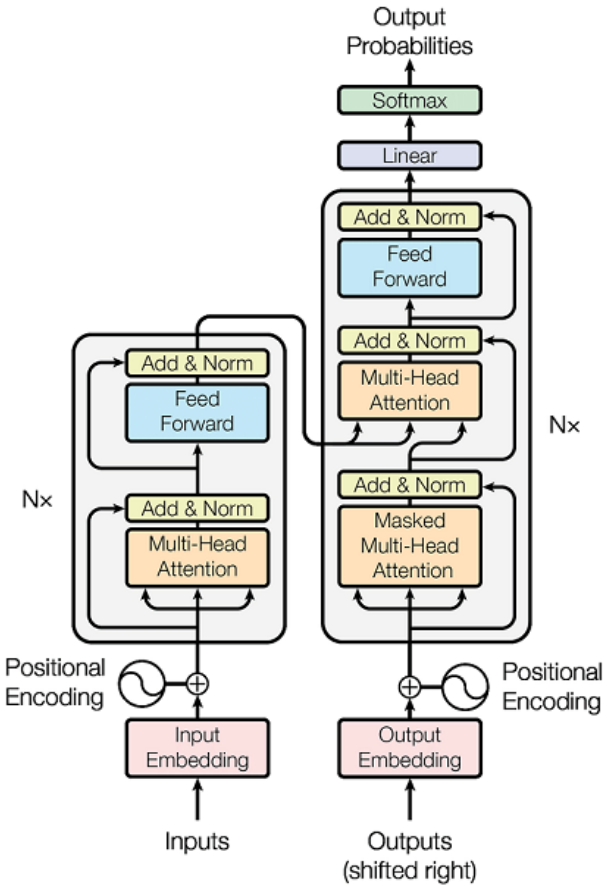


Figure 2.6: Architecture of the Transformer model (extracted from [2])

- In the ”encoder-decoder attention” layers the queries come from the previous decoder layer, and the keys and values come from the output of the encoder, allowing every position in the encoder to attend over all positions in the input sequence.
- The self-attention layer in the encoder gets its keys, values and queries from the output of the previous layer in the encoder, with each position in the encoder having the ability to attend all positions in the previous layer of the encoder.
- In the decoder, the self-attention layers allow each position in the decoder to attend all positions up to (and including) that position. As already mentioned, this is made through the use of masking inside the attention function.

Also, because the model does not contain any recurrence or convolution, there is a need to inject information about the relative or absolute position of the tokens in the sequence. Those are called positional encodings and are added to the input embeddings at the bottom of both encoder and decoder stacks. The users decided to use sine and cosine functions of different frequencies:

$$\begin{aligned} PE_(pos, 2i) &= \sin(pos/10000^{2i/d_{model}}) \\ PE_(pos, 2i+1) &= \cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{2.3}$$

where *pos* is the position and *i* is the dimension with $0 \leq i \leq \frac{d_{model}}{2}$, meaning that each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. The authors chose this function because they "hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset *k*, PE_{pos+k} can be represented as a linear function of PE_{pos}".

In the final step of the architecture, the decoder stack outputs a vector of floats, that go through a final linear transformation (layer), followed by a softmax layer. The Linear layer is a fully connected neural network that projects the vector of stacked decoders into a larger vector called a logits vector, that later enters the softmax layer and transforms those scores into probabilities. Each cell has a probability (as they all add up to 1), so the cell with the highest probability is chosen, resulting in the word associated with it being the prediction output for that time step [58].

2.3 Chapter summary

This chapter describes the AKI problem and gives an introduction to attention mechanisms and Transformers. Regarding AKI, besides the complications associated with the disease, there's an overview of the most used staging systems and the most common baselines adopted. On the related work, studies addressing AKI and other clinical tasks were mentioned. The studies tackling other diseases were acknowledged for displaying several methodologies that can be applied to AKI tasks.

The section tackling attention mechanisms gives an introduction to the concepts of encoder-decoder, Transformers and self-attention. Encoder-decoder is directly related to Transformers, as it is used in its architectures to collect global dependencies between inputs and outputs, while completely discarding the use of any recurrence that was used in former attention mechanisms. Transformers use self-attention across different locations in its architecture, and were at the basis of the model architecture used in this thesis.

Chapter 3

Self-Attention Model

With the purpose of creating the first attention based sequence modeling architecture for multivariate time-series data, in 2017, Song et al. [3] took inspiration in the Transformer model by Vaswani et al.[2]. Solely relying on self-attention mechanisms, the Simply Attend and Diagnose (SAnD) architecture, named by the authors, discards any recurrence or convolutions for sequence modeling, as they have computational restraints, such as the inability to be trained in parallel.

The Transformer [2] follows the overall architecture of an encoder-decoder structure. Since attention mechanisms have produced good results on transduction tasks in NLP - which in machine translation indicates the production of sequences of words in a target language given examples in a source language -, and specifically self-attention has been used successfully in a variety of NLP tasks, as seen in the latest section, the authors approach focused on studying their effectiveness in clinical diagnosis for several tasks. In this chapter, the several components of the SAnD architecture will be explained. Some implementation details regarding the model will be in the end of the chapter.

3.1 Architecture

The Transformer architecture [2] receives a sequence of symbol representations $(x_1, x_2, \ldots, x_T)$, such as words when performing on machine translation benchmarks, later transformed into a continuous representation $\mathbf{z}$ by the encoder, followed by the decoder, that produces the output sequence of symbols $(y_1, y_2, \ldots, y_T)$. In the context of working with EHR/clinical data, the input sequence for SAnD [3] is a sequence of clinical measurements $(x_1, x_2, \ldots, x_T)$, $x_t \in \mathbb{R}^R$ where R denotes the number of variables, with the objective of generating a sequence-level prediction for any specific task, as it can be denoted as a discrete scalar y for multi-class classification, a discrete vector $\mathbf{y}$ for multi-label classification or a continuous value y for regression problems.

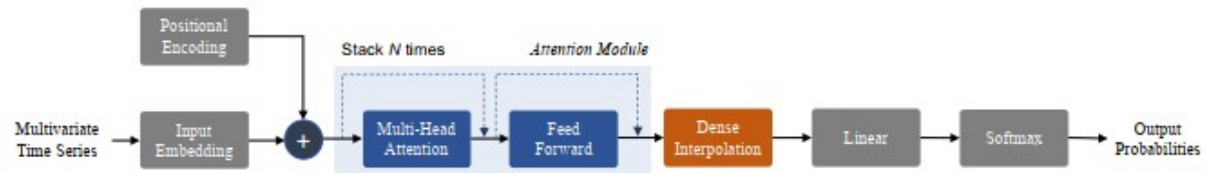


Figure 3.1: Overview of the proposed approach for clinical time-series analysis (designed by Song et al.[3]).

Just like in the Transformer scheme, the attention module consists in N identical layers, with each

one containing the attention mechanism and a feed-forward sub-layer, along with residue connections.

3.1.1 Input Embedding

The first step of this architecture is similar to the input embedding step in most NLP architectures, where the mapping of words from a sentence originates a high-dimensional vector space that helps facilitate the actual sequence modeling [59]. Given the R measurements at every time step t, an embedding that captures the dependencies across different variables is generated, without considering any temporal information. A 1D convolutional layer with kernel size 1 is employed to obtain the d-dimensional ($d > R$) embeddings for each t. Denoting the convolution filter coefficients as $\mathbf{w} \in \mathbb{R}^{T \times h}$, where h is the kernel size, we obtain the input embedding: $\mathbf{w} \cdot \mathbf{x}_{i:i+h-1}$ for the measurement position i.

3.1.2 Positional Encoding

As addressed in Section 2.2.2, each word in a sentence goes through the Transformer's encoder/decoder stack, with the model itself not having any sense of position for each word.

The authors way of fixing the lack of information about the order of the sequence was through the addition of $positional\ encodings$ to the input embeddings of the sequence, which provides information on the relative or absolute position of the time-steps in the sequence. In this work specifically, this encoding is achieved through mapping time step t to the same randomized lookup table during both training and prediction. The d-dimensional positional embedding are then added to the input embedding with the same dimension. As the authors mentioned, using sinusoidal functions like the original Transformer [2] is an alternative to this approach.

3.1.3 Attention Module

As said before, $SAnD$'s architecture focuses almost entirely on self-attention mechanisms. In detail, it's used a restricted self-attention that imposes causality, meaning it only considers information from previous positions where the analysis is occurring.

Self-attention is designed to capture dependencies of a single sequence, and in this model, the range of dependency is a specified parameter that indicates how far the attention model can look into the past in order to obtain the representation for each position. This is named as masked self-attention by the authors, and is important due to different tasks requiring longer range dependencies than others, i.e phenotyping tasks require longer range dependencies compared to mortality prediction [3], both used in their study.

In general, as mentioned in Section 2.2.2, an attention function can be defined as mapping a query $\mathbf{q}$ and a set of key-value pairs $\mathbf{k}$, $\mathbf{v}$ to an output $\mathbf{o}$. For each position t, the attention weighting is computed as the inner product between $\mathbf{q_t}$ and keys at every other position in the sequence (within the restricted set) $\{\mathbf{k'_t}\}_{t'=t-r}^{t-1}$, where r is the mask size. Using these attention weights,$\mathbf{o}$ is computed as weighted combination of the value vectors $\{\mathbf{v'_t}\}_{t'=t-r}^{t-1}$ and pass $\mathbf{o}$ through a feed-forward network to obtain the vector representation for t. Mathematically, it can be expressed as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d}}\right)\mathbf{V} \tag{3.1}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the matrices formed by query, key and value vectors respectively, and d is the dimension of the key vectors. This mechanism is the same scalar dot-product attention used in the original Transformer architecture. Since only self-attention is used, $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ all correspond to input embeddings of the sequence (with position encoding). Additionally, as said before, the sequence is masked to specify how far the attention models can look into the past for obtaining the representation for each position.

Also, this architecture also uses the multi-head attention similar to the Transformer. With the purpose of creating multiple attention graphs 8 heads are used, and the resulting weighted representations are concatenated and linearly projected to obtain the final representation. As the authors explain: "implicitly, self-attention creates a graph structure for the sequence, where edges indicate the temporal dependencies. Instead of computing a single attention graph, we can actually create multiple attention graphs each of which is defined by different parameters. Each of these attention graphs can be interpreted to encode different types of edges and hence can provide complementary information about different types of dependencies". The second component in the attention module is 1D convolutional sub-layers with kernel size 1, similar to the input embedding. Internally, two 1D convolutional sub-layers are used with ReLU (rectified linear unit) activation in between. Besides that, residue connections are included in both the sub-layers. Since the attention module is stacked N times, the actual prediction task occurs using representations obtained at the final attention module.

3.1.4 Dense Interpolation for Encoding Order

Unlike transduction tasks, predictions are not made at each time step in all cases. Consequently, there is a need to create a concise representation for the entire sequence using the learned representations, which is done through the use of a dense interpolated embedding scheme, that encodes partial temporal ordering.

The simplest approach to obtain a unified representation for a sequence, while preserving order, is to simply concatenate embeddings at every time step. However, in this case, it can lead to a very high-dimensional, "cursed" representation which is not suitable for learning and inference. Instead of simply using the embeddings at each time step to make predictions, just like what happens on transduction tasks, a concatenation of all the embeddings for each time step would be required in this case. Dense interpolation prevents that high-dimensional representation for a sequence, compressing the embedding at every step into a single vector representation. In this architecture, the pairing of dense interpolation embeddings with the positional encoding module, are highly effective in capturing enough temporal structure required to challenge clinical prediction tasks.

This means that embeddings outputted from the multi-headed-attention module are taken and used in a manner that is useful for capturing syntactic and structural information. As demonstrated by Russell et al. [60], dense interpolated embeddings not only provides a concise representation (of the sequence), but also found that encoded word structures are more useful in detecting syntactic features.

The pseudo-code to perform dense interpolation for a given sequence is shown in Figure 3.2. Denoting the hidden representation at time t, from the attention model, as $\mathrm{s_t} \in \mathbb{R}^d$, the interpolated embedding vector will have dimension $d \times M$, where M is the *dense interpolation factor*. Note that when $M = T$, it reduces to the concatenation case. The main idea of this scheme is to determine weights w, denoting the contribution of $\mathrm{s_t}$ to the position m of the final vector representation u. As the algorithm iterates through the timesteps of a sequence, s is obtained, the relative position of time step t in the final representation u and w is computed as $w = \left(1 - \frac{|s-m|}{M}\right)^2$. In Figure 3.3 there's a view of the dense interpolation process in an example with $T = 5$; $M = 3$. The larger weights in w are indicated by darker edges while the lighter

Dense Interpolation Embedding
Input : Steps t of the time series and length of the sequence T, embeddings at step t as $\mathbf{s}_t$, factor M.
Output: Dense interpolated vector representation $\mathbf{u}$.
for $t = 1$ *to* T **do**
  $s = M * t/T$
  for $m = 1$ *to* M **do**
    $w = pow(1 - abs(s - m)/M, 2)$
    $\mathbf{u}_m = \mathbf{u}_m + w * \mathbf{s}_t$
  end
end

Figure 3.2: Dense interpolation embedding with partial order for a given sequence

edges indicates lesser influence. In practice, dense interpolation is implemented efficiently by caching w's into a matrix $\mathbf{W} \in \mathbb{R}^{T \times M}$ and then performing the following matrix multiplication: $\mathbf{U} = \mathbf{S} \times \mathbf{W}$, where $\mathbf{S} = [\mathrm{s}_1, \ldots, \mathrm{s}_\mathrm{t}]$. Finally, $\mathbf{u}$ is obtained as a result of stacking columns of $\mathbf{U}$.

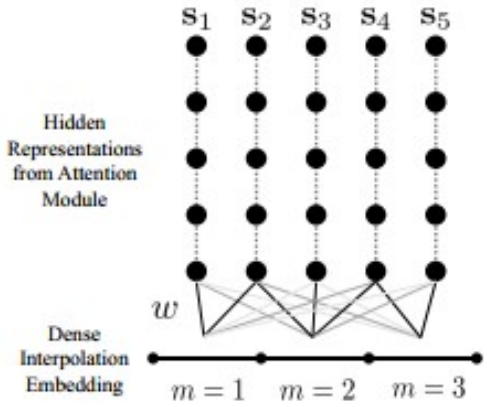


Figure 3.3: Visualization of the dense interpolation module, when T = 5 and M = 3

3.1.5 Linear and Softmax layers

In the end, just like in the Transformer architecture, the single vector representation acquired before is fed through a linear layer to obtain the logits, using a linear layer, which feeds a specific layer depending on the task. Here, the authors used a softmax layer for the binary classification problems, a sigmoid layer for multi-label classification since the classes are not mutually exclusive and a ReLU layer for regression problems.

In this study, the decision was to use Pytorch's *CrossEntropyLoss*[61], which is a combination of the library's *LogSoftmax* and *NLLLoss* (negative log likelihood loss), suggested when working on classification problems with C classes (where $C > 2$). The loss function is:

$$\begin{aligned} \mathrm{loss}(x, class) &= -\log\left(\frac{\exp(x[class])}{\sum_j \exp(s[j])}\right) \\ &= -x[class] + \log\left(\exp(s[j])\right) \end{aligned} \tag{3.2}$$

Also, it's possible to assign weight to each of the classes, which is useful when working with an unbalanced training set, that seems to be the case in this work, as will be addressed in Section 5.3.

3.1.6 Regularization

During training, the regularization on this architecture happens at the sub-layer level, and in the attention weights of the module. Regarding the former, dropout [62] is applied to the output of each sub-layer in the attention module prior to residual connections, followed by a normalization of the outputs. Also, an additional dropout layer is included after the addition of the positional encoding to the input embeddings. Besides the regularization in the layers during the forward pass, it's also performed attention dropout, similar to the Transformer architecture, after computing the self-attention weights.

3.1.7 Complexity

Working with long-range dependencies is a tough task for many sequence modeling tasks, and that challenge is well expressed by the computational complexity revolving around them. One notion of complexity is the amount of computation that can be parallelized, measured as the minimum number of sequential operations required. Recurrent models require $O(T)$ sequential operations with a total $O(T \cdot d^2)$ computations in each layer. In comparison, the proposed approach requires a constant $O(1)$ sequential operations (entirely parallelizable) with a total $O(T \cdot r \cdot d)$ computations per layer, where r denotes the size of the mask for self-attention. In all implementations the mask size will be far from the value of d, meaning that $r \ll d$. As a result of that, it's possible to see that this approach is significantly faster than using RNN training.

3.2 Implementation details

Due to the original code from the article not being shared publicly, there was a need to use an unofficial adaptation from the original model, developed by Hirotaka Kawashima [63]. Consequently, compared to SAnD, this architecture varies in small details, i.e. its implementation does not use the same Positional Encoding method, because in the article there was no detailed description of random lookup tables at all. So, as an alternative approach, Kawashima used the Transformer-style Positional Encoding used in Vaswani et al.'s [2] work. As said before, its architecture used sinusoidal functions to produce a sense of order in the sequence, providing information about the position of each element in the sequence through the inclusion of the positional encoding on top of the actual embeddings. In the end, this alteration to the model shouldn't change much the final results, because the goal of position encoding is to teach the model information about time, which is met by both methods. The sinusoidal functions are:

$$\begin{aligned} PE_(pos, 2i) &= \sin(pos/10000^{2i/d_{model}}) \\ PE_(pos, 2i+1) &= \cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{3.3}$$

As said before in Chapter 3.1.5, the loss function used in this approach is different than the ones used in the original article, and the decision was to use pytorch's CrossEntropyLoss.

With the purpose of better tracking the model results, its hyperparameters, and also taking into account the capacity of reproducibility from the models, Comet.ml was used. It is a platform created to ”provide[s] insights and data to build better, more accurate AI models while improving productivity, collaboration and visibility across teams” [64]. Using Comet.ml makes it specially easier to compare results, while also evidencing the hyperparameters/metrics for each instance. The usage of Comet.ml was already implemented in the code by Khirotaka, but some functions were added in order to automatically calculate the metrics:

- Sensitivity (Recall);
- Specificity;
- Precision;
- F-score;
- ROC Curves and their respective AUC scores.

Chapter 4

Data

4.1 MIMIC-III

In this study, the data used is from the MIMIC-III database [40], an acronym for the Medical Information Mart for Intensive Care, a large database with information regarding de-identified ICU patients admitted from 2001 to 2012 to the Beth Israel Deaconess Medical Center (BIDMC) in Boston, Massachusetts. MIMIC-III is a public database, available on the PhysioNet website [65], only subject to a formally required request in order to maintain the appropriate care and respect to the detailed information it contains. The request includes completing an online human-subjects training course, and the signing of a data use agreement, allowing an unrestricted data analysis upon acceptance.

There has been a concerted move towards the adoption of digital health record systems in hospitals in the recent years. In the US, for example, the number of non-federal acute care hospitals with basic digital systems increased from 9.4 to 75.5% over the 7 year period between 2008 and 2014 [66]. Despite this advance, interoperability of digital systems remains an open issue, leading to challenges in data integration. As a result, the potential that hospital data offers in terms of understanding and improving care is yet to be fully realized. In parallel, the lack of reproducibility of studies is increasingly coming under criticism within the scientific research community [67].

Knowing these issues, that's where MIMIC-III are set to make a difference and elevate the research community. MIMIC-III supports a diverse range of analytic studies spanning epidemiology, clinical decision-rule improvement, and electronic tool development. Being freely available to researchers worldwide, owning a diverse and very large population of ICU patients in which containing highly granular data, including vital signs, laboratory results, and medications. Also, due to the increasing usage of the previous major releases, MIMIC-III is expected to be widely used internationally in academic and industrial research areas. Thus, the open nature of the data allows clinical studies to be reproduced and improved in ways that would not be possible otherwise.

The data itself consists in over 40,000 distinct patients, with information sparse through 26 tables, including demographics, vital sign measurements made at the bedside ($\sim$1 data point per hour), laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality in the ICU and after being discharged (if it happened)[40, 65]. The patients in the database can be divided in 2 groups based on their age, with patients under 1 year old labelled as neonates, and the rest as adults, starting at the age of 14. The MIMIC-III database was populated with data that had been acquired during routine hospital care, so there was no associated burden on caregivers and no interference with their workflow.

Before data was incorporated into the MIMIC-III database, it was first deidentified in accordance with Health Insurance Portability and Accountability Act (HIPAA) standards using structured data cleansing and date shifting. The deidentification process for structured data required the removal of all eighteen of the identifying data elements listed in HIPAA, including fields such as patient name, telephone number, address, and dates. In particular, dates were shifted into the future by a random offset for each individual patient in a consistent manner to preserve intervals, resulting in stays which occur sometime between the years 2100 and 2200. Time of day, day of the week, and approximate seasonality were conserved during date shifting. Dates of birth for patients aged over 89 were shifted to obscure their true age and comply with HIPAA regulations: these patients appear in the database with ages of over 300 years.

Despite all that, as explained in Correia et al.'s study [17], one of the problems of dealing with MIMIC-III was the fact that data is not consistently available for all the patients, due to it being originated from two different systems, CareVue and MetaVision, where the same measure can have many different codes, complicating the task of reproducing the time series with all of the variables. In her study, a semi-manual identification of the repeated data was made, checking each measurement code, which also turned out useful on merging values with different measuring units to the same unit. This shows the importance of the data preprocessing task, which will be addressed in the next section, requiring attention to several details regarding the different measures and labels of each variable.

4.1.1 Table Selection

As said before, MIMIC-III is a relational database containing tables of data relating to patients who stayed in the ICU. Out of the 26 tables, MIMIC-III gathers detailed information of patient's stays scattered across 6 of them, such as information regarding hospital admissions, ICU stays, patient demographics and details about the clinical services, as well as its location within the hospital. During the exclusion criteria in section 4.2.1, there was a need to combine several of these tables to have information about, for example, the first ICU stay for patients with more than one ICU stay.

The selected data used in this study was only collected on the tables CHARTEVENTS and LABEVENTS, as they're the two biggest tables regarding time series type of data from the patient in the ICU. The former contains all the charted data available for a patient across it's ICU stay, while the latter contains information regarding laboratory based measurements, not only including measurements from in-hospital scenarios, but also out of hospital laboratory measurements from other clinics, thus referred to as "outpatients". In this work, only data collected within a patient's ICU stay is considered, so every outpatient data was discarded.

Also, the Urine Output records used in the staging criteria are collected within another table - OUTPUTEVENTS, which contains information on fluids that have been excreted by the patient -, but because those collected values were only used in the staging criteria, and the UO features later used in this work are calculated using records from this table, their origin in the list of features in Appendix A will be labelled as calculated.

Linking the patient's information tables with their associated dictionary table was also needed to obtain the label for every measurement during all this process.

4.2 Data Preprocessing

In this section, some choices were based on the work from Correia et al.[17], mainly due to the positive results obtained in her work. Both studies have different goals and approaches, which is why the usage of this study as groundwork will only turn out useful for some of the sections, more specifically, in the sections involving data preprocessing.

4.2.1 Exclusion Criteria

MIMIC-III only contains information of neonates, children below 1 year old, and patients over 15 years old. Just like in Correia et al.'s study, this study will also focus on adults, making it necessary to remove the patients whose age is below 14. For that, the age is calculated using the difference between the earliest admission data and the date of birth (DOB), except for patients older than 89 years old - that show up in the database as 300 years old - due to being considered a vulnerable group of patients. Their DOB is shifted to mask their age and comply with HIPAA [68].

As said before, because there is no specific feature information regarding AKI development during the ICU stay, there is a need to calculate the stage of the disease regularly, based on the patient's physiologic information. For that, Correia et al. decided to use the AKIN classification system, following the criteria addressed in the sections before (See Table 2.1), except for about 35% of patients with lack of data on UO. In those cases, the AKI stage was only calculated using the SCr values. Those particular choices will be different in this study, besides the choice to use the KDIGO classification system, and will be addressed in the section 4.2.4.

Despite that variation, other common exclusion criteria amongst the studies included the removal of patients that did not stay at least 24 hours in the ICU, had a minimum of three measures of SCr and one measure of UO at every 6 hours. Also, for the patients that were admitted more than once in the ICU, only the first event of AKI was considered, avoiding biased assessments. Discarding patients with no Weight records was also needed, because otherwise it wouldn't be possible to calculate the UO staging criteria. The flowchart of that process is in Figure 4.1.

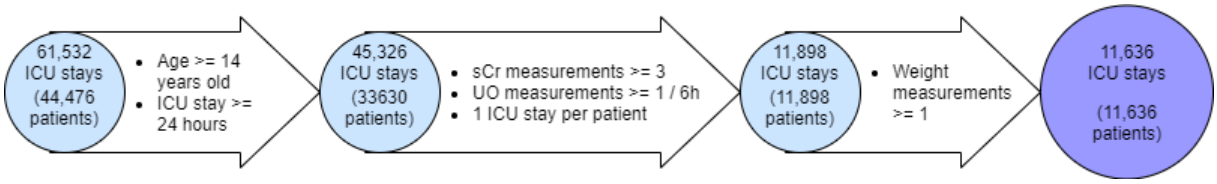


Figure 4.1: Flowchart of the number of patients in the cohort when applying the exclusion criteria

After applying all the stipulated criteria for the patients, the next step was to collect all the data from the tables mentioned before and proceed to clean it. By this time, 11636 patients remain in the cohort with a total of 571 different ID codes. Each ID code represents a measure for one of the two information systems. Most of them do not have a clear label name, making it difficult to understand the exact measure for every Item ID (ID code). Besides that, within the same information system several measures are sparse through different ID codes, as they're collected from different sensors. For a better way to deal with those different ID codes and label them correctly, two similar data extraction resources shared by the MIMIC-III community were combined to get the maximum variables possible [69, 70].

Those consist in groupings of Item ID's into variables, with also information regarding the normal value ranges for every variable extracted.

Using the two benchmarks referenced before, 410 of the 571 Item ID's were grouped into 82 different variables. The rest of them were manually dealt with, some were labelled and included in the variables when easily identifiable or deleted if they were duplicates or unidentifiable. After this long process, 147 variables were identified and processed.

Also, the following features were added to the feature list, directly derived from the KDIGO criteria used to calculate the AKI stage:

- From the SCr criteria:
  - SCr;
  - Lowest SCr value from the last 48 hours;
  - Lowest SCr value from the last 7 days.
- Because the UO criteria is calculated through the rate, using a sum of several sensors (sum of different Item ID's related to UO), these were the obtained variables:
  - UO rate for the last 6 hours;
  - UO rate for the last 12 hours;
  - UO rate for the last 24 hours.

Cleaning each feature/code values was another procedure that had to be done manually. Not only removing errors and null values, but the categorical data also needed to be dealt with. Categorical variables were handled in a way to make them ordinal, ordering them by symptoms severity, e.g., Urine Appearance had 5 different labels and was ordered as: Clear = 1 , Cloudy = 2 , Sediment = 3 , Sludge = 4, Clots = 5. Another particular example is the Glasgow Coma Scale (GCS), a practical scale described as a way to communicate about the level of consciousness of patients with an acute brain injury [71]. The GCS total score has values between 3-15 and is achieved with the sum of the behavioral response of the patient's eye movement, verbal and motor response. Besides GCS total score, identified as a numerical feature, the three behavioral responses are also used in this work as categorical features. All the categorical features labels and values can be seen in the Appendix A, where each feature is identified by their origin, with the features derived by the KDIGO criteria identified as calculated features.

Regarding discrete features, race and gender will be discarded from the features. Age will also not be included in the features used in the prediction, but will be used for patient stratification. Patient's height will also not be taken into account, as the priority was to keep features whose values change through time.

In this work, the choices made were based on an attempt to keep the coherence on initial goals, such as to recreate, by the point of view of a health specialist, the prediction of the patient's health situation simply looking at its feature measures. Besides that, the self-attention model has the capacity to analyse the sequence over time in a feature, so there was no win scenario in having fully simulated values for an entire feature in a patient. To fight that, the decision here was to only work with patients that have values for every feature, which consequently tends to reduce significantly the cohort with the increasing number of features selected. The decision was to choose the first 80 features present in more patients, and further keep only the patients with values in all of those selected features, and as expected, the number of patients was drastically removed to 393.

Next step was to remove values from every feature that was out of the normal value range, as well as null values. As said before, the features created using both data extracting resources had the normal value ranges for every variable, which means that the variables that were manually grouped had not. Those normal range values were decided based on looking at the histogram of values for every variable. Also, there was more than one different unit of measure in the records for both the selected variables and the originated from the community benchmarks, so there was a need to understand what were the most coherent units of measure to use between the Conventional and the SI Units [72]

Keeping only the measurements within each feature's normal range meant some features got reduced to the point where there were not values in every patient. Thus, the number of patients and features got reduced to 375 and 64, respectively (including the calculated features derived from the KDIGO criteria). This whole process is shown in Figure 4.2

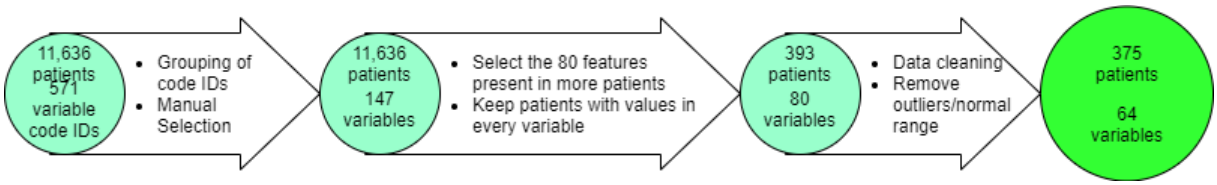


Figure 4.2: Flowchart of the data preprocessing process with the number of patients and features throughout

When in doubt of the relevance of some features during this whole process, particularly the variables that were manually selected, finding studies that used them to study AKI (or other renal disease) was the trigger to decide whether to keep them or not, e.g Basophils, in which Mack & Rosenkranz [73] studied the affect on immune responses and discussed implications for renal diseases. Other variable examples that required studies or articles in order to be selected were Base Excess (arterial and venous) [74], Calcium and Creatine Phosphokinase (CPK) [75], Central Venous Pressure (CVP) [76], Urine color and appearance [77] and Ectopy Frequency [78]. Some features were selected based on their presence in studies on AKI and were later discarded, due to the lack of records after the data cleaning, e.g. Vancomycin, an antibiotic often associated with nephrotoxicity, was tested on its direct responsibility in the occurrence of vancomycin-associated nephrotoxicity (VANT) for fragile patients with multiple risk factors for AKI in Filippone et al.'s study [79]. Other examples were Nitric Oxide [80], Serum Osmolality [81] and Serum Uric Acid [82].

4.2.2 Repeated data

As already addressed in the Data section (Section 4.1), MIMIC-III requires an identification of the repeated data, due to it having several different codes for the same measures. Each measurement or concept has a specific and unique code called Item ID (e.g. Item ID = 211, describes measurements of heart rate (HR)). Nonetheless, there are duplicated codes for each concept (e.g. HR has two codes assigned: Item ID = 211 and Item ID = 220045), associated from the fact that the information comes from two distinct critical care information systems and also because of the free text nature of data entry in the older system.

The process to counter that started with text search to look for codes with identical names, then investigate whether the selected codes were measuring the same things, and later checking the range of values through the use of box plots for each code, to see if the range of values was similar, and consequently comparable.

The benchmark created by Yereva et al. [69] deals with the same measurement for both information systems. The variables originated from it were not changed, just the manually selected features were required to have a code identification process with the purpose of collecting the most amount of information possible. Also, the benchmark has the normal range of values for each of their recognized variables, meaning that the particular process of understanding the range of values was not needed for every Item ID, thus being only used in the codes and variables manually selected.

4.2.3 Baseline Estimations

Regarding the baseline of serum creatinine, this study will use the lowest value of the first three measures of SCr, basing this choice on the studies that used the same data set (MIMIC-III) [16, 17, 18], as seen in the Related Work section (Section 2.1.3). The SCr records used for the staging were only from the ID code 50912, from the LABEVENTS table, just like the research community uses in the MIMIC code repository [83]. The remaining records for other codes associated to SCr will be used in the final dataset as the feature Serum Creatinine (scr).

Both the UO and the SCr baseline values were generated with scripts from the MIMIC code repository [83], and followed the criteria from the KDIGO classification system shown in Table 2.1, which has the criteria regarding SCr and UO. In this work, RRT initiation and Anuria were not used in the AKI classification.

4.2.4 Classification systems

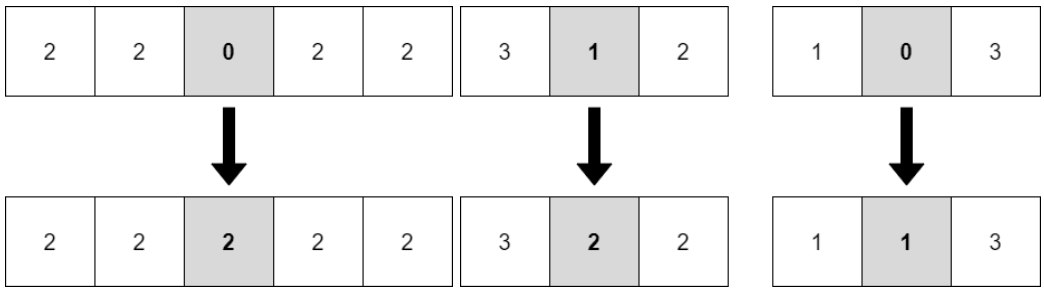
The limitation of information regarding the measurements used for the staging classification (SCr and UO) was addressed in Correia et al.'s study [17], using mainly SCr values to calculate the AKI stage (when both weren't available), not discarding the patients when the UO data was not enough to be useful on the AKI staging. Different from what was done in Tomašev et al's study [42], as the authors only worked with stages originated from SCr values. In this thesis, two distinct classification systems will be tested: one using only SCr values to calculate the AKI stage, and the other that uses both SCr an UO criteria. For the latter, this means that for each record of either SCr or UO, there is an AKI stage calculated. Thus, and because both AKI staging criteria are not associated by any means, some values might be in disagreement. Both measures have different registry frequencies, with UO generally being recorded hourly and SCr more in a random pattern, meaning that in the space of an hour the predicted AKI stage can change meaningfully.

In order to fight that, some alterations were made in the full timeline of the AKI staging criteria, looking for a more coherent sequence. Those alterations were made only when the discording values were originally from different classification systems, and are illustrated in Figure 4.3:

- The discording value was between 4 values identical from the same classification system. In that case, the discording value was changed to be the same as those other 4 (Figure 4.3a);
- The values before and after were different, and the discording value also different from both of them. In that case, the lowest of the neighbours is taken (Figure 4.3b).

To facilitate the flow of the sentences the classification systems used will be abbreviated to:

- sCr - classification system that only uses SCr values for the evaluation;



(a) Example 1 (b) Example 2

Figure 4.3: Examples of the creation of 2B

- 2B Raw - classification system with the SCr and UO evaluation, following the criteria;
- 2B - classification system generated through the alterations from 2B Raw with the purpose of providing coherence within the sequences.

This creation of 2B was not developed with the purpose of being groundbreaking, it was mainly to replicate the way a specialist would look at the sequence and evaluate the patient. Assuming the raw output of the sequence (2B Raw) is the real AKI development in a patient, it wouldn't be legitimate to bounce between having no AKI (stage 0) to, for example, stage 2, within the space of an hour. This example is displayed in Figure 4.3a.

In section 5, there's a result comparison between 2B and 2B Raw, to test and evaluate any differences in the prediction results between them. This means that there will be 3 different classification systems considered for the next section, with the purpose of also using them in the final results.

4.2.5 Missing Data Imputation

One important task regarding the data was the need of reshaping the full dataset in a way that would fit the model. Due to SAnD not dealing with any missing values, and because the goal was to specifically work with data in hourly time stamps, reshaping was necessary to facilitate the selection of sequences for the predictions. Also, because not every patient has values for each feature in every hour, either because the regular extracting time does not happen on an hourly basis for every feature, or due to the data cleansing process removing some records, missing data imputation occurred.

A function that runs through the whole dataset was created, with each record having an identification for the patient, the feature and the time it was recorded in the format of hours since admission. In the first step of the function, a complete representation for each patient in a table like way is generated. The number of rows for each patient's table depends on the length of its stay in the ICU, while the columns are the features. Having that, all values are inserted into the table using the hour, H, and the feature, F, like coordinates, (H, F). There were some cases where more than 1 record was recorded in the same hour for the same feature, thus the following procedure took place:

- If those values were duplicates, keep only one of them;
- For the categorical features, the last record is the set value for the hour;
- For the numerical features, the set value is the average of those records;

- When there was no record for the previous/next hour, the first/last value filled it. When both are missing, both are completed with the same 'rule', while the value for the original hour is set as the last recorded value.

As said before, because there are not values for every feature in every hour, the following methods were applied:

- Last observation carried forward (LOCF) to fill in missing values from left to right, until another value is recorded or until the end of the sequence;
- Next observation carried backward (NOCB) to complete the missing values that take place before the first value for a feature is recorded.

This process is shown in Figure 4.4, where in Figure 4.4a is an example of the table creation for a patient with the input of all its available measures, and Figure 4.4b displays the application of LOCF and NOCB, indicated with blue and red, respectively.

	Features				
	Oxygen Saturation	Potassium	Heart Rate	Hemoglobin	Calcium
1	97		75		
2			76	10.9	7.7
3		6			8.5
4	92		83		

(a) Table with all patient values

	Features				
	Oxygen Saturation	Potassium	Heart Rate	Hemoglobin	Calcium
1	97	6	75	10.9	7.7
2	97	6	76	10.9	7.7
3	97	6	76	10.9	8.5
4	92	6	83	10.9	8.5

(b) Application of the LOCF and NOCB methods

Figure 4.4: Example of the input missing data process for a patient

Here, the methods used could be different, e.g, using interpolation to input the missing data, specially on discrete data, but the main idea, again, was to complete the patient table in a way comparable to what a specialist would see. Since SAnD takes into account the variance of values for every feature during the full training sequence, different interpolation methods would probably payoff in better results, which will be addressed in the future work section, on section 7.

The function also generates the data sequences, so one last parameter of the function created is the length of the training sequences. Thus, after each patient's table is fully completed, each table results in $t - l$ training sequences with size l, with t being the number of hours in the table for that patient. This process is best illustrated in Figure 4.5.

4.2.6 Descriptive Statistics of the selected patients

From the 375 patients cohort, there's a clear tendency for patients closer to elderly ages. Only 60 from the total patients are of age 50 or below (16%), while 173 (about 46.2 %) are 70 or older. This comes as no surprise, due to the usual tendency of the elderly to stay longer in the ICU, meaning there was more

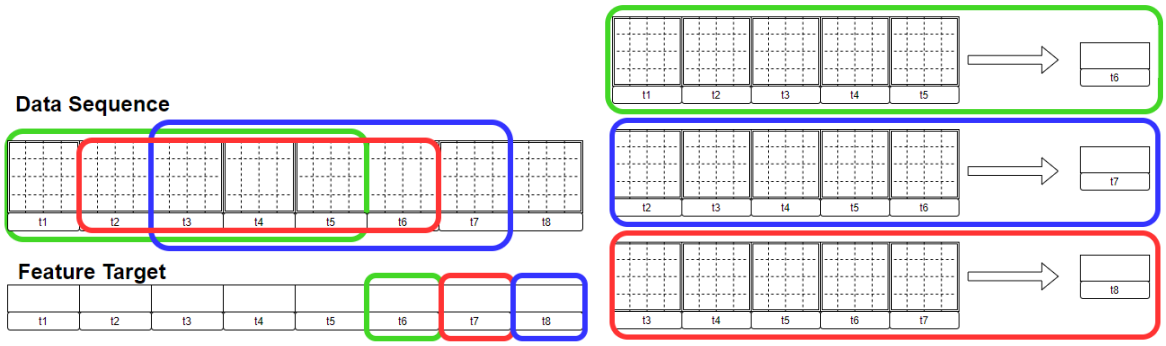


Figure 4.5: Example of training sequences extracted out of an 8 hour sequence, with a length of 5 hours

chances of being measured for every feature considered, which was the main requirement of the patient selection, as mentioned in the previous sections. By looking at the length of stay average (in days) per age in Figure 4.6, with the context of the reduced number of patients in younger ages, means that the few younger patients in the cohort were very specific cases of patients that stayed for longer in the ICU.

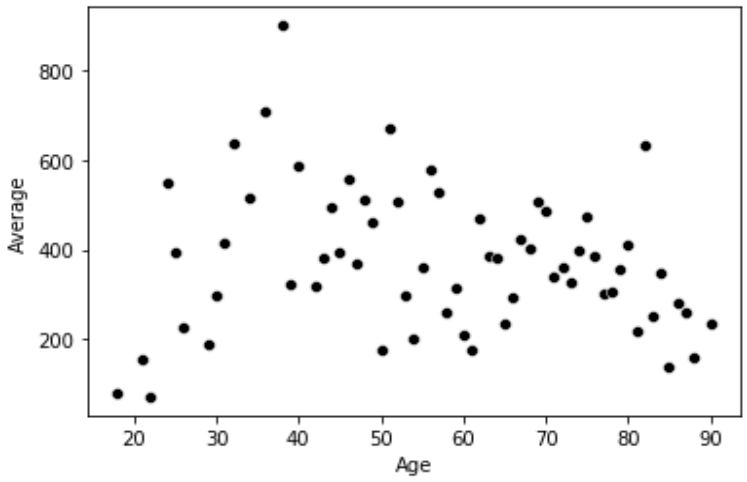
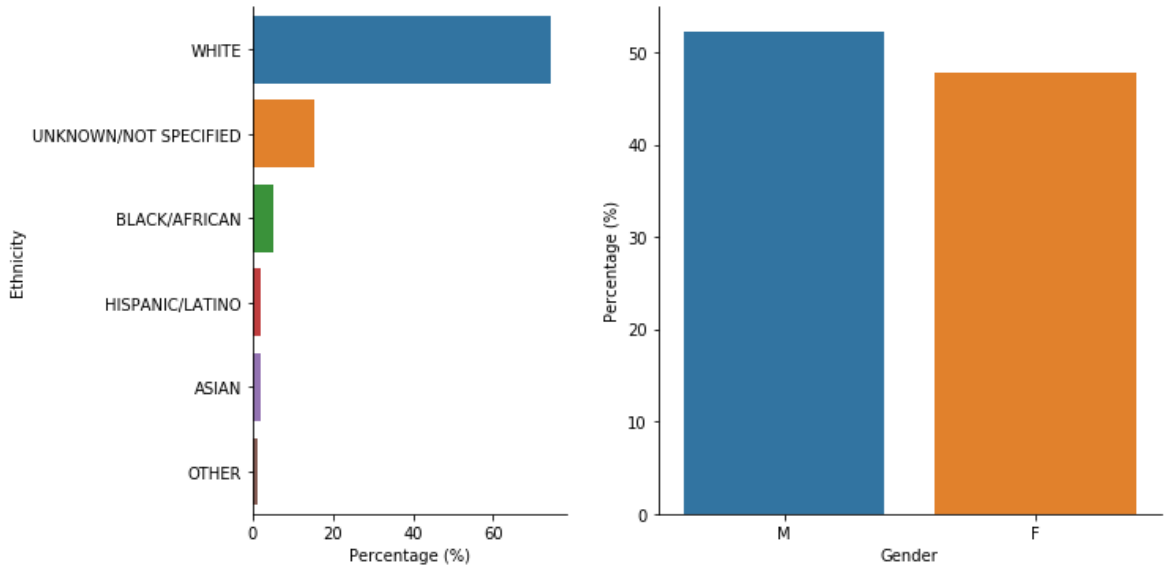


Figure 4.6: LOS average (in days) per age

The patient distribution by age shows that it is possible to stratify the patients and study the age factor. It is harder to infer the statistical relevance of race since the cohort was very imbalanced, with 279 out of the 375 patients being white (74.4%) and only 19 black (5.1%), in contrast to the 57 not specified (15.2%), the remaining patients were 8 latino, 7 asian and 5 other ethnicity non specified (2.1%, 1.9% and 1.3% respectively). In terms of gender the cohort was better balanced, males represent the majority with 52.3%, counting 196 against the 179 females in the patients taken into account in this study.

4.2.7 Time Windows

As addressed in the related work on other diseases (Section 2.1.4), where the studies analysed focused on using time windows with the purpose of building their predictive models [43, 42], the same will happen in this study, where the goal is to utilize supervised learning based on time windows to predict the progression of AKI, learning through patients with consistent information on those time windows.



(a) Race distribution (b) Gender distribution

Figure 4.7: Demographics from the patient cohort

The approach in this study is quite similar to the approach by Tomašev et al. [42] in the sense of working with continuous time windows, with information for every hour. Several sequence lengths will be used, this way we'll be capable of studying the constant progression of the patients and test what sequence lengths end up producing better results. The length of sequences used will be 6h, 12h and 24h. A higher sequence length won't be addressed because one of the exclusion criteria was to keep patients that have been at least 24h in the ICU, thus some of the patients in the final cohort didn't stay in the ICU for much longer than that.

For every patient, this approach will continue as far as there is data capable of doing so, meaning that probably the prediction will only stop when the patient dies, or is discharged.

4.2.8 Patient Stratification by age

Due to the general performance of predictive models not being uniform across the entire population, knowing the differences across the different subgroups can be important regarding future practicalities on those patients. Having that in mind, and based on the approach by Tomašev et al. [42], evaluating the model performance across different clinical subgroups can be interesting. Those different subgroups were created based on the patient demographics, more specifically through age groups.

The subgroups selected were:

- Patients aged below 40 (Young);
- Patients aged between 40 and 60 (Middle age);
- Patients aged between 60 and 80 (Older age);
- Patients aged over 80 (Elder).

As can be seen in Table 4.1, the class imbalance is evident. Thus, we will not study and compare the progression of the disease along the subgroups. The low percentage of some stages will not enable

			Stage distribution (%)			
	Number of patients	Sample size	Stage 0	Stage 1	Stage 2	Stage 3
Young	23	9523	61.58	23.12	9.78	5.52
Middle age	94	36851	83.62	6.60	5.90	3.88
Older age	177	60798	76.37	14.74	6.90	1.99
Elder	81	22348	73.89	19.13	3.97	3.01

Table 4.1: Sample size of the target class for each age group, using the sCr classification system with 24h sequences

making legitimate comparisons, meaning there won't be any valuable conclusions as the results on those stages will be considerably worse.

4.3 Chapter summary

In this chapter, the full data prepocessing pipeline was described, including the several procedures and the thoughts behind each choice made. In the end, the final cohort consists of 375 patients with information regarding 64 features (including the calculated features, the group of features originated from the criteria used to define the patient's AKI stage), as the list of features can be seen in the Appendix A.

The data collected from the patients was then processed into sequences with hourly measures for every feature. These sequences will be used in the final section, as we will test the results when using sequences of 6h, 12h and 24h length.

The AKI classification systems chosen to work with were addressed and explained: 2B, 2B Raw and sCr. Later, in the next sections, there will be a comparison between them based on their results.

Chapter 5

Feature Importance using Random Forest

In this section we'll focus on analyzing the individual importance for every feature when predicting the patient's AKI Stage. Using the preprocessed data, each hourly timestamp was used as training data for the two types of predictions that were made: predicting the AKI stage on the current exact hour and predicting the AKI stage of the following hour. Within those predictions, several experiments with different sets of features occurred. We tested predicting with and without the features originated from the classification systems, to examine the performance of the remaining features. Despite using sequences with several hours in the model, the results obtained in this section withstand as legitimate because the whole point of predicting the current and the next point in time was to analyze the possible differences or similarities of the feature importance results between them. If a feature is seen as important while predicting both cases, there's a reasonable assumption that it would also be an important feature to use in predictions when it's fed into a sequence based model.

5.1 Feature Importance

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable [84]. Those scores play an important role in a predictive modeling project, including providing insight into the data, better understanding the model, and the basis for dimensionality reduction and feature selection, that can improve the efficiency and effectiveness of a predictive model on the problem.

Feature importance scores can be calculated for problems that either involve predicting a numerical value or problems that involve predicting a class label - which is this case-, called regression and classification respectively.

Most importance scores are calculated by a predictive model that has been fit on the dataset and can provide insight into it, as the relative scores can highlight which features may be most relevant to the target, and contrarily, which features are the least relevant to the model when making a prediction. Knowing what features to select or keep can simplify the problem that is being modeled, speed up the modeling process (working with less features means less complexity), and in some cases, improve the performance of the model. This is a type of model interpretation that can be performed for those models that support it.

There are several types and sources of feature importance scores, although popular examples include statistical correlation scores, coefficients calculated as part of linear models, decision trees, and permutation importance scores.

In this work, scores were calculated using decision trees, more specifically through fitting the dataset into a Random Forest model with 100 trees. Permutation Importance was also considered in the initial testing stage for this section, but because the results were not conclusive of anything different from the Feature Importance the initial plan using both was discarded.

The goal here is not to achieve some ground breaking results such as finding a new feature that helps the AKI stage prediction in a general way. The idea is just to understand what are the significant features for this specific dataset, and compare the model's performance when using variable selection by focusing only on the important features.

With knowledge of the significant features for every case, the accuracy of predicting the AKI Stage in the current hour and the next was assigned into Random Forest classifiers (RF) and Naive Bayes classifiers (NB). Those classifiers were fed the dataset with a reduced number of features, using only the x most important features - with x being the predefined number of features to select and test -, taking into account the importance scores order achieved before. This testing started with the 5 most important features and went all the way up to the point of including all features, this way it's possible to compare the accuracy of both models when dealing with different numbers of features. Thus, knowing the optimal number of features to use in order to obtain good results means there is less testing to do with the final model, as we can exclude testing results with irrelevant features. Random Forest with 100 number of trees and Gaussian Naive bayes with scikit learn's default $var\_smoothing$ value of $1e^{-9}$ were used.

5.2 Experiments using all patients

In this section, the cohort for all patients considered is the cohort of 375 patients drawn in the previous data preprocessing segment. As said before, some tests were made in this section, and besides analyzing the results when predicting the current and next hour, it was also tested the results with the following selections of features:

- In the current hour:
  - All features (excluding AKI Stage);
  - All features (excluding AKI Stage and the calculated features).
- In the following hour:
  - All features;
  - All features (excluding AKI Stage);
  - All features (excluding AKI Stage and the calculated features).

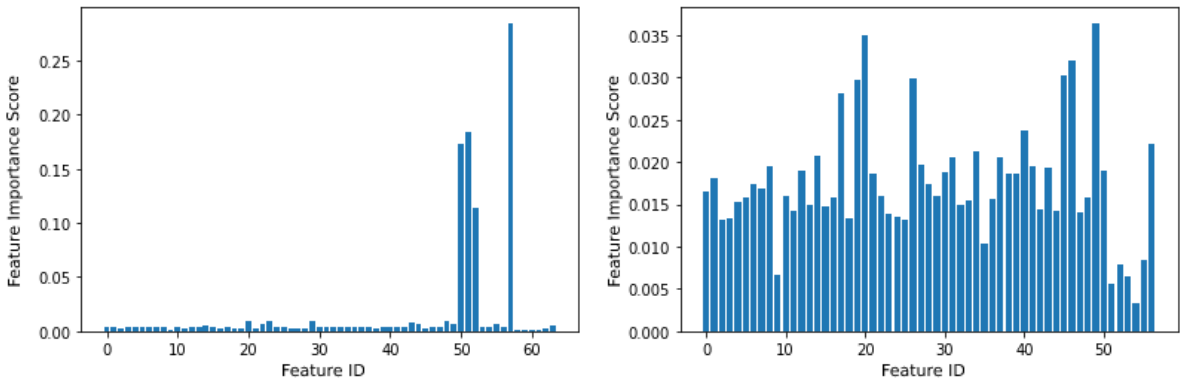
5.2.1 Feature Importance

The feature importance scores between current and following hour prediction were very similar, with the slight exception of some features who got small variations and because of that changed their positions in the ordered feature ranking, but because those importance scores were so low those changes of order were not significant. Thus, only the scores of the following hour predictions will be analyzed here.

As expected, when using all features the most important feature on the 3 classification systems is AKI Stage, followed by the 3 calculated UO features on both 2B and 2B Raw, or followed by the 3 calculated sCr features on the sCr classification system. The remaining features have close scores, with very low values (figures 5.1 and 5.2). Because the results in this section were similar between 2B and 2B Raw, only the results for 2B will be displayed, but the Feature Importance scores for all classification systems considered can be seen in the Appendix B.

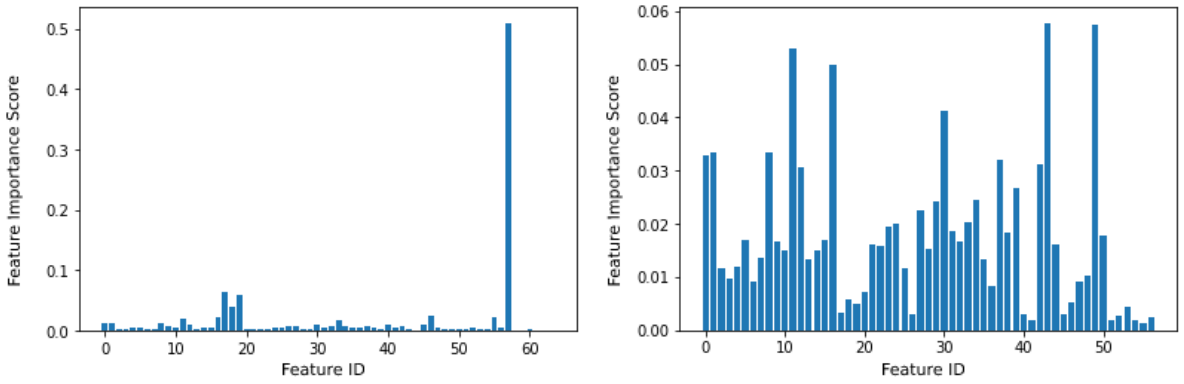
Although the classification systems use different criteria, some of the features with the highest feature importance scores were common to all the prediction models, such as Weight, Prothrombin Time (PT) or Platelets. Without the calculated features, all these remaining scores were residual with no standout values, as the highest score out of the 3 classification systems was Serum Creatinine (sCr) with 0.057 in the sCr classification system.

At the bottom of these importance score rankings were the categorical features. Out of those, only Urine Color got a better placement in 2B, in the sCr classification system none of those were significant to the prediction models. This may happen due to the tendency of this method to favor numerical features and categorical features with high cardinality [85].



(a) Using all features (b) Using all features except the calculated features

Figure 5.1: FI scores for the 2B classification system when predicting the following hour



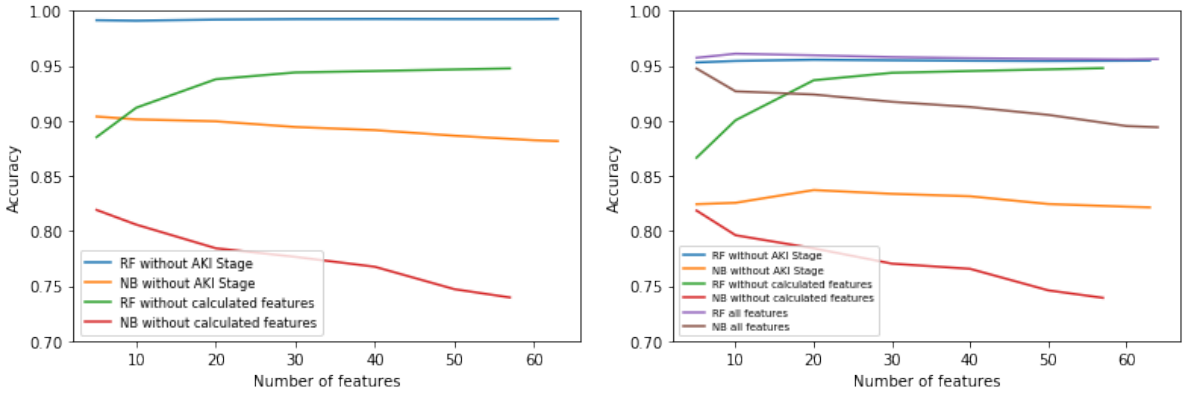
(a) Using all features (b) Using all features except the calculated features

Figure 5.2: FI scores for the sCr classification system when predicting the following hour

5.2.2 Model efficiency using different number of features

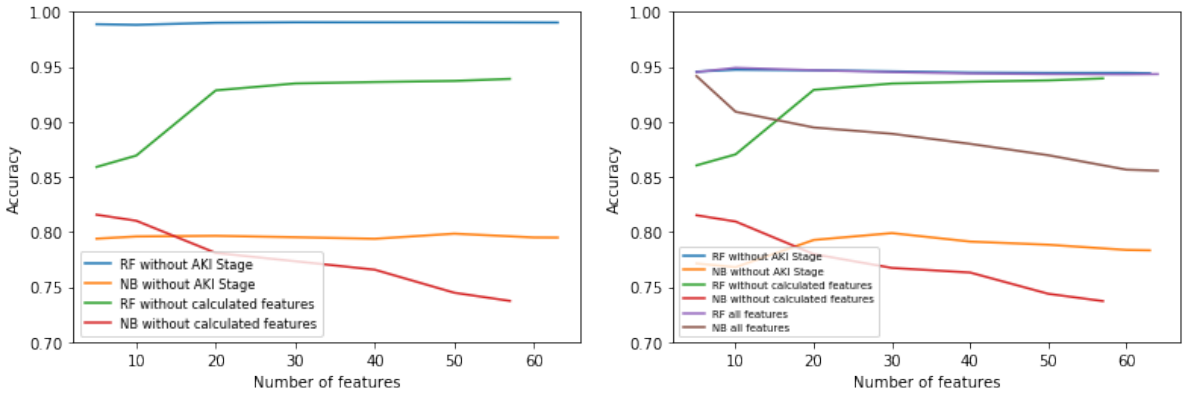
The next step was to actually test the results when using those different set of features. Here, the features were selected based on their respective importance scores, from best to worst. The results used in the figures 5.3 to 5.5 are in the appendix D in a table format.

The results were generally better when using a Random Forest classifier compared to using Naive Bayes, as can be seen in the figures 5.3 to 5.5. As expected, the overall accuracy of the prediction models decreases when the calculated features are not included. Also, looking at the figures it's possible to visualize that the results were progressively worse as the number of features included in the model were higher. The only exception were the RF models with no calculated features, as they increased nearly 10% in accuracy using all features in the set, compared to using the 5 most important ones.



(a) Accuracy when predicting the current hour (b) Accuracy when predicting the following hour

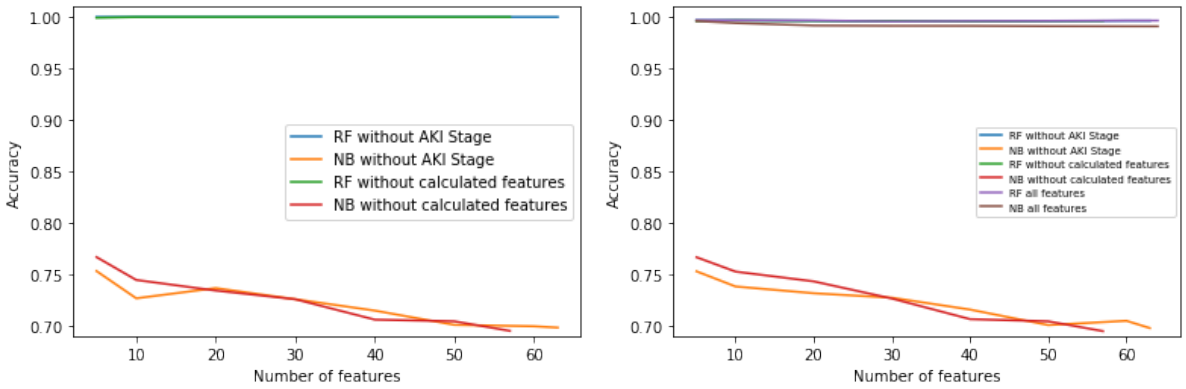
Figure 5.3: Using 2B to evaluate the predictions, while also showing its accuracy for different sets of features selected



(a) Accuracy when predicting the current hour (b) Accuracy when predicting the following hour

Figure 5.4: Using 2B Raw classification system to evaluate the predictions, while also showing its accuracy for different sets of features selected

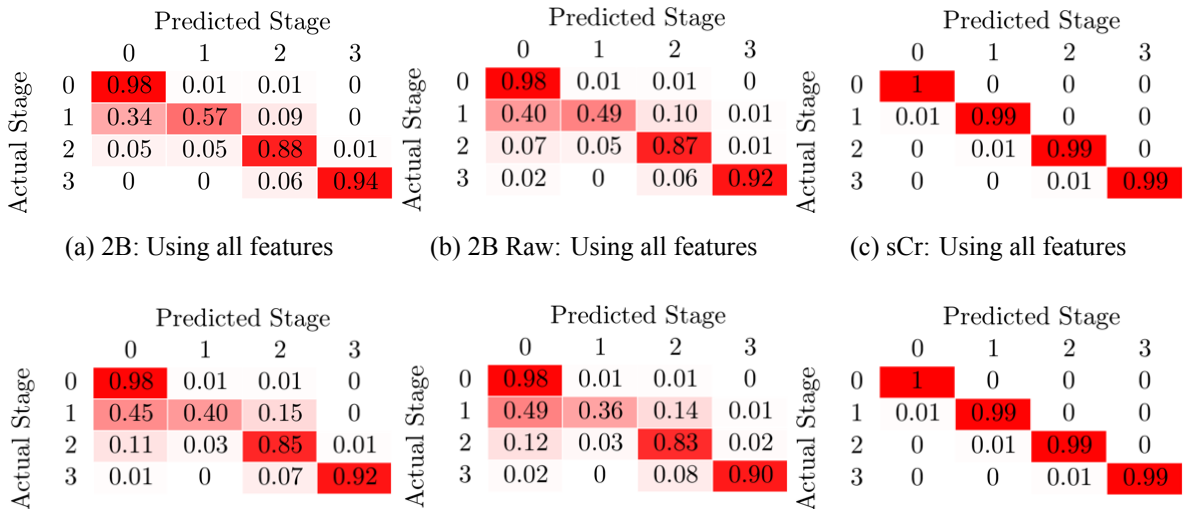
Despite the good results in general, specially using RF models, more important than the overall accuracy was knowing the efficiency when predicting every class. By looking at the confusion matrices from the predictions made (with all features) for the following hour (Figures 5.6 and 5.7), it's clear that there is a struggle to correctly predict stages 1, 2 and 3, which can be explained by the imbalance of the



(a) Accuracy when predicting the current hour (b) Accuracy when predicting the following hour

Figure 5.5: Using sCr classification system to evaluate the predictions, while also showing its accuracy for different sets of features selected

classes, and will be addressed in the next section. Still, 2B shows slightly better results both in RF and NB, compared to 2B Raw, while both have low accuracy on predicting stage 1 occurrences. When using all stages, results for the sCr classification system are nearly immaculate, and stays that way across all RF predictions. Across the 3 classification systems using the NB classifier, sCr does predict stages 1, 2 and 3 with a higher precision, and as expected, the accuracy of predicting those 3 stages is worse when excluding the calculated features.



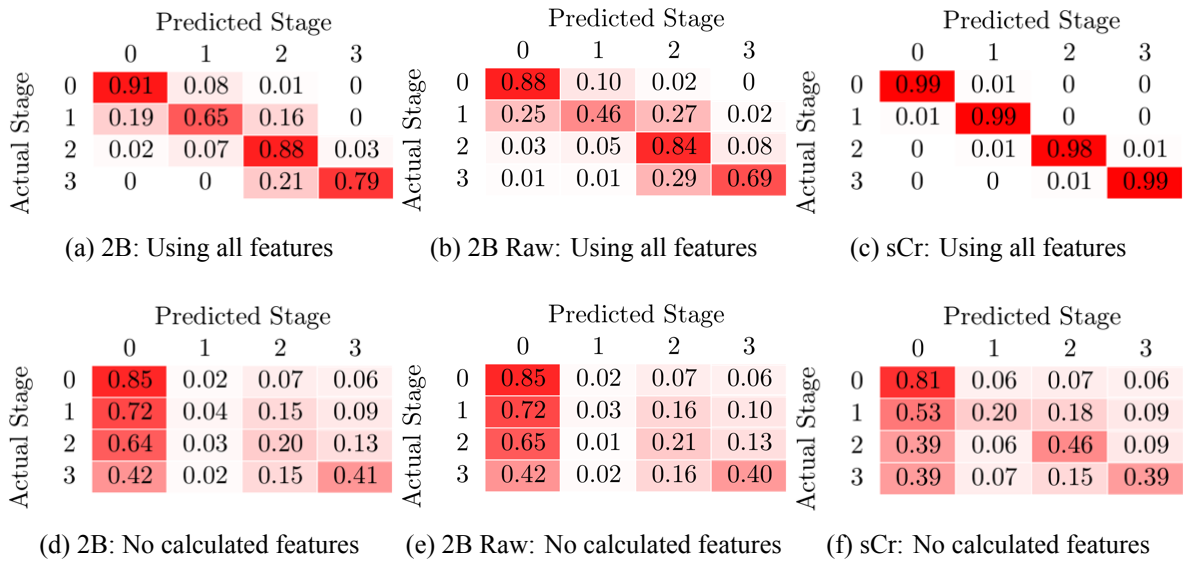
(a) 2B: Using all features (b) 2B Raw: Using all features (c) sCr: Using all features

(d) 2B: No calculated features (e) 2B Raw: No calculated features (f) sCr: No calculated features

Figure 5.6: Confusion matrices when predicting the following hour with RF

5.3 Dealing with class imbalance

As mentioned in the former section, the general low accuracy when predicting AKI stage 1, 2 and 3 was alarming and needed to be addressed. The AKI stage classification values for every hour is counted and presented in Table 5.1, where the high class imbalance stands out, for both types of classification



(a) 2B: Using all features (b) 2B Raw: Using all features (c) sCr: Using all features

(d) 2B: No calculated features (e) 2B Raw: No calculated features (f) sCr: No calculated features

Figure 5.7: Confusion matrices when predicting the following hour with NB

		Stage			
	Number of patients	0 (%)	1 (%)	2 (%)	3 (%)
2B	375	83	4	10	3
2B Raw					
sCr		78	13	5	4

Table 5.1: Class imbalance for all patients selected

systems used. When using the SCr classification system from KDIGO, the frequency of each stage was approximately 78%, 13%, 5% and 4%, from stage 0 to stage 3 respectively, and approximately 83%, 4%, 10% and 3% when using both 2B classification systems. This shows the discrepancy between classes and helps understand the difficulties to predict the less frequent stages. Despite the good results in terms of overall accuracy, specially using the calculated features, testing the outcome when using training data with more balanced classes is important, specifically to understand the capability to correctly predict the remaining stages.

Originally, the process of balancing the classes was thought out in two different ways: either remove hourly records without looking at the patients and get the perfect balance of 25% for every class, or remove the patients with more recordings of stage 0 and try to balance the classes the best way possible. The latter option was taken into account, because later we'll be working with sequences from the same patients, so this same patient selection can be further used. Patients were removed (in a descending way) by the proportion of stage 0 in the total of stage values, until the frequency of stage 0 values drops to at least 50%, preferably stopping before removing patients with a reasonable amount of stages 2 and/or 3.

This process had to be done individually for each classification systems, and in the end, the final class frequency turned out to be approximately 50%, 8%, 29% and 13% (from stage 0 to 3, respectively) for both 2B and 2B raw, and 30%, 26%, 26% and 18% for the sCr classification system (Table 5.2).

		Stage			
	Number of patients	0 (%)	1 (%)	2 (%)	3 (%)
2B	53	49	8	29	14
2B Raw	61	50	8	29	13
sCr	66	30	26	26	18

Table 5.2: Class imbalance after removing patients

5.4 Experiments with balanced classes

5.4.1 Feature Importance

Comparing to the former section, when looking at the Feature Importance using the reduced patient cohort (Appendix C) a slight alteration is visible in terms of orders of the features, in general, but also detail that those feature scores did not actually change significantly. This means that the non-calculated features continue to show a lack of significance on the predictions, exhibiting the lack of individual importance that these features actually have in the predictions.

Both in 2B and in the sCr classification systems the scores did not change too much, thus some features topped the list with the reduced cohort as well. In the sCr classification system, Serum Creatinine, CPK and Weight were the most relevant features in common between both cohorts, while in the 2B and 2B raw Heart Rate, Temperature, Systolic Blood Pressure and Mean Blood Pressure stayed relevant for both cohorts, with, once again, equal feature importance scores.

The same output happened within current hour and following hour prediction, where the scores remained pretty much identical, meaning that an assumption can be made that these most relevant features will stay being relevant independently of the point in time of the prediction, so we will assume that each important feature will be important in general.

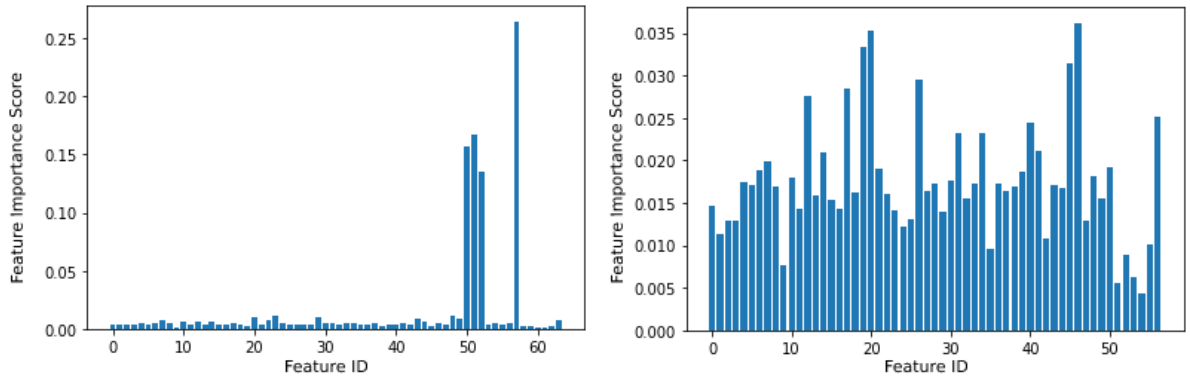
The only difference between cohorts is that the score of AKI Stage decreased from approximately 0.51 to 0.33, which indicates that AKI Stage has less weight on the final prediction when predicting the following hour. In 2B and 2B Raw AKI Stage also lost importance, but the scores only decreased approximately 0.02 on both.

As expected, the categorical variables had low feature importance scores, just like what happened before, with exception of Urine Color that actually got a good placement in the list on the 2B and 2B raw classification system.

These results show that besides the calculated features, there are no features that have significant relevance in the prediction. Thus, there's no point in trying the final model with feature sets without those calculated features, as it is expected to produce worse results without them.

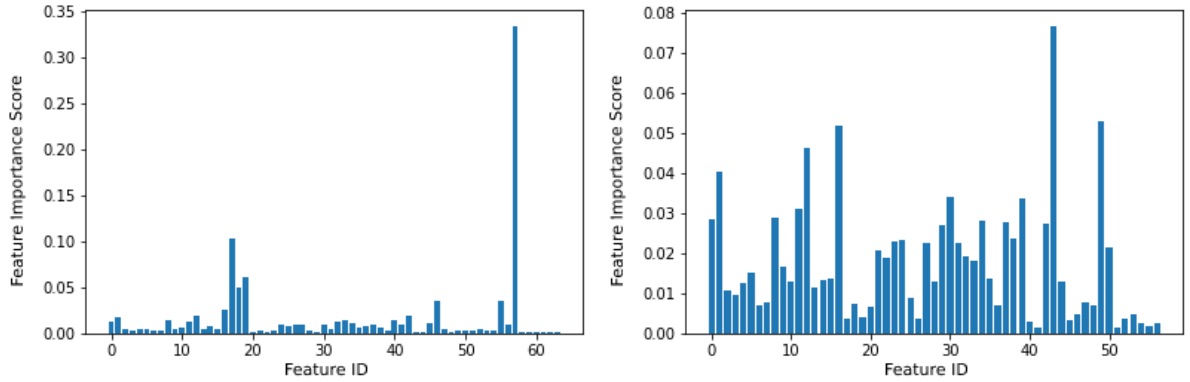
5.4.2 Model efficiency using different number of features

Once again, the results used in the figures 5.10 to 5.12 are in the appendix E in a table format. No significant changes occurred when comparing to the cohort with all patients. Again, the results were generally better when using RF compared to NB, the overall accuracy of the prediction models decreased when the calculated features were not included. Also, with this reduced cohort the results continued to get progressively worse as the number of features included in the model got higher.



(a) Using all features (b) Using all features except the calculated features

Figure 5.8: FI scores for the 2B classification system when predicting the following hour using the reduced cohort



(a) Using all features (b) Using all features except the calculated features

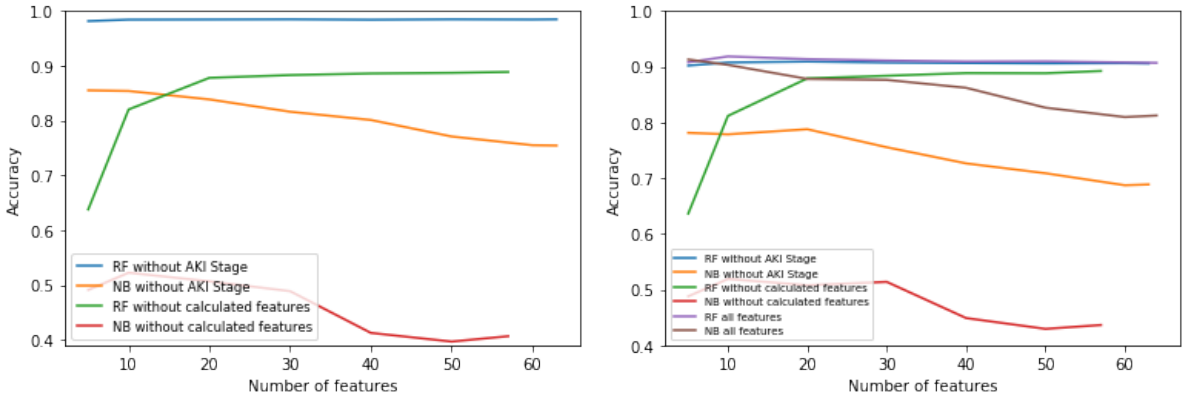
Figure 5.9: FI scores for the sCr classification system when predicting the following hour using the reduced cohort

The accuracy of correctly predicting stage 1, 2 and 3 was better in this reduced cohort, in pretty much every case for every classification system. Stage 1 was where the biggest improvements happened, showing the importance of balancing the stages to improve the accuracy in every feature. Again, 2B shows slightly better results both in RF and NB, compared to 2B Raw, but not close to the results that the sCr classification system got (Figures 5.13 and 5.14).

5.5 Final conclusions

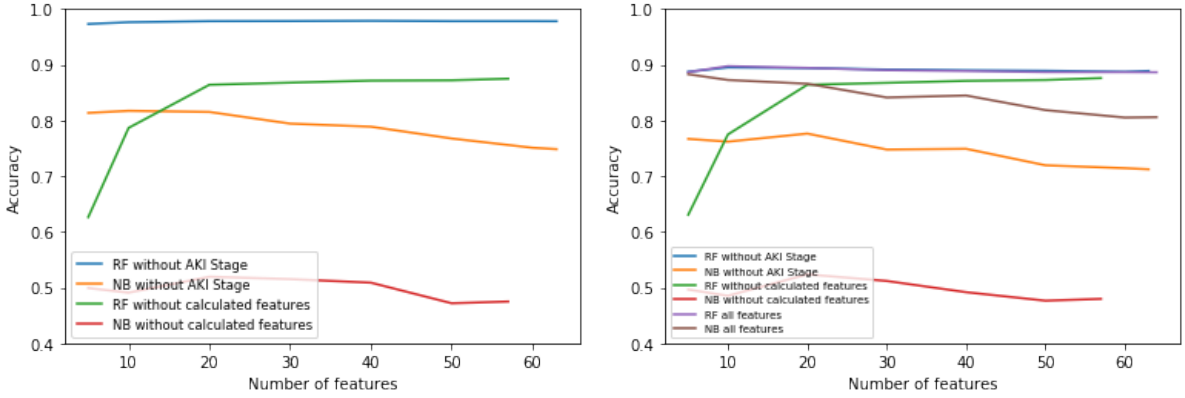
Across all of this section it's possible to see that the sCr classification systems produces better results than both 2B and 2B raw. So it is expected to have better accuracy not only generally, but also better accuracy predicting stage 1, 2 and 3.

Reducing the number of patients and consequently balancing the stage proportions within the cohort produced the results we wanted to see. Both using RF and NB, the accuracy of predicting the stages with AKI occurrence (stages 1, 2 or 3) increased when predicting the 2B and 2B raw classification systems, although there was still some trouble predicting stage 1, which can be explained by the low proportion



(a) Accuracy when predicting the current hour (b) Accuracy when predicting the following hour

Figure 5.10: Using 2B with the reduced cohort to evaluate the predictions, while also showing its accuracy for different sets of features selected



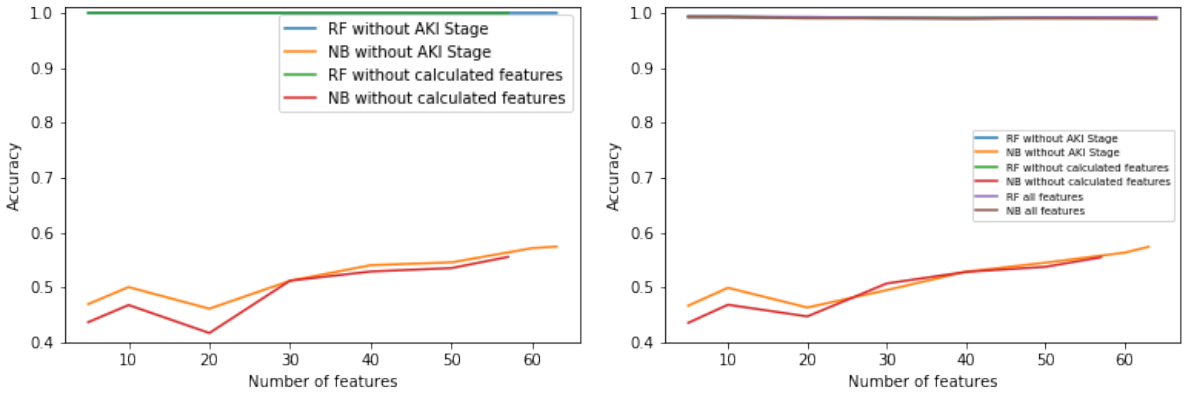
(a) Accuracy when predicting the current hour (b) Accuracy when predicting the following hour

Figure 5.11: Using 2B Raw with the reduced cohort to evaluate the predictions, while also showing its accuracy for different sets of features selected

of stage 1 values in the model. For NB, the accuracy of stages with AKI occurrence after removing the calculated features is really low, and, in general, RF resulted in better results compared to NB. The predictions of the sCr classification system barely changed, as it already had accuracy close to 1 in every stage.

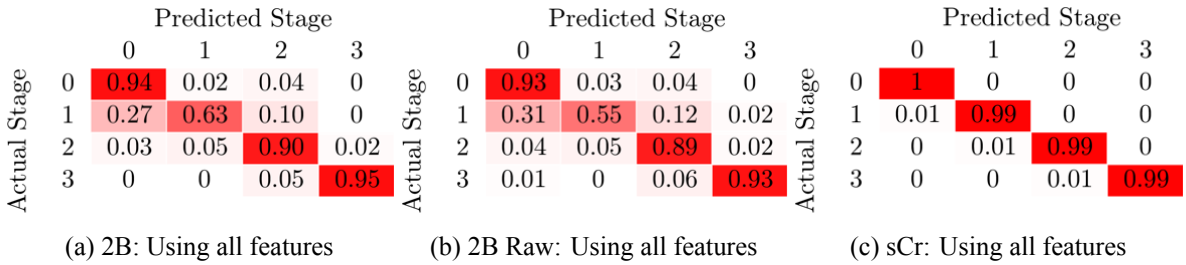
As said before, this section was also used to compare 2B and 2B raw, and exclude the one with worst results of the final model tests. Using NB produced better results on 2B in every set of features tested, while RF drew better results for 2B only when using all features. In the other cases there was a slight increase in 2B raw, although not anything significant, but because the calculated features are going to be included in the final model tests, the assumption that 2B will produce better results than 2B Raw can be made, as it outperformed in almost every cases.

Having that, the decision to keep on using 2B on the final section was made, and consequently excluding 2B raw from now on. Also, because the reduced patients cohort produced better results in terms of accuracy in predicting stages with AKI occurrence, it was decided that the tests in the next section will only happen using those reduced cohorts. No clear evidence suggested that a high number of features should be used when working with SAnD. So, we'll consider the optimal number of features to use is 10.

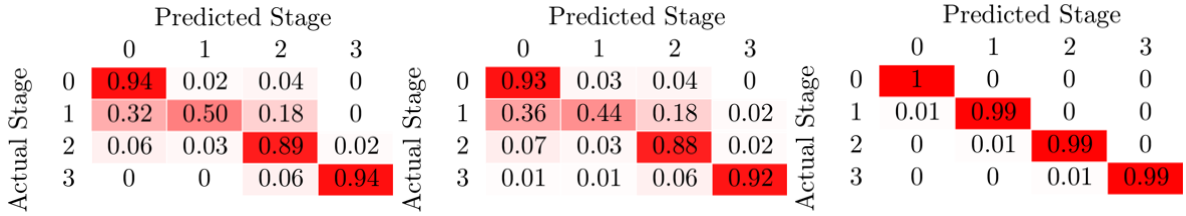


(a) Accuracy when predicting the current hour (b) Accuracy when predicting the following hour

Figure 5.12: Using sCr with the reduced cohort to evaluate the predictions, while also showing its accuracy for different sets of features selected



(a) 2B: Using all features (b) 2B Raw: Using all features (c) sCr: Using all features



(d) 2B: No calculated features (e) 2B Raw: No calculated features (f) sCr: No calculated features

Figure 5.13: Confusion matrices when predicting the following hour with RF

(a) 2B: Using all features (b) 2B Raw: Using all features (c) sCr: Using all features

(d) 2B: No calculated features (e) 2B Raw: No calculated features (f) sCr: No calculated features

Figure 5.14: Confusion matrices when predicting the following hour with NB on the reduced cohort

Chapter 6

Results

6.1 Model Assessment

6.1.1 Classification metrics

The real output data (the target data) is non-binary, since the AKI stage can vary from 0 to 3, depending in the severity of the disease. This means that we have a 4 by 4 confusion matrix, and because the majority of classification metrics are defined for binary cases we can break down this multiclass problem into several binary ones. The prediction of each AKI stage will be a different task, such as:

- AKI stage 0: stage 0 vs. not stage 0 (stage 1, 2 and 3)
- AKI stage 1: stage 1 vs. not stage 1 (stage 0, 2 and 3)
- AKI stage 2: stage 2 vs. not stage 2 (stage 0, 1 and 3)
- AKI stage 3: stage 3 vs. not stage 3 (stage 0, 1 and 2)

This way, the predicted outputs and the target values can be plotted in a binary confusion matrix. Having that, the performance metrics used in this work are:

Overall Accuracy - ratio between the number of correct predictions and the total number of predictions made.

Specificity - ratio between the number of correct negative predictions and the total number of negative points. It gives the power of the model to correctly predict other stages than the AKI stage in matter.

$$Specificity = \frac{TN}{TN + FP} \tag{6.1}$$

Sensitivity (Recall) - the ratio between the number of correct positive predictions within the total number of real positive points. It gives the power of the model to predict the exact AKI stage among those stage predictions.

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

Precision - the ratio between the number of correct positive predictions within all positive predicted points. It gives the proportion of actual positive AKI stage predictions are correctly classified within all positive predictions.

	Number of patients	Sequence length	Sample size	Stage distribution (%)			
				Stage 0	Stage 1	Stage 2	Stage 3
sCr	66	6	27672	29.54	26.01	26.16	18.29
		12	27276	28.64	26.27	26.54	18.55
		24	26484	27.33	26.32	27.27	19.08
2B	53	6	18032	48.60	8.11	29.57	13.72
		12	17714	48.71	7.97	29.37	13.95
		24	17078	49.49	8.02	28.79	13.70

Table 6.1: Sample size for the reduced cohorts

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

F-score (F_1 score) - harmonic mean of precision and recall. If either precision or recall have low values the F_1 score will suffer and result in a lower score. This means that a high F_1 score indicates low rates of false positives and false negatives.

$$F_1 score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6.4}$$

AUC - area under the receiver-operating characteristic (ROC) curve gives a good assessment of the balance between sensitivity and specificity, two important scores when dealing with imbalanced datasets. It is a performance rate that is easily understandable by caretakers, and is frequently used in medical literature, making it possible to compare methods and models easily.

6.1.2 Sample size and stage distribution

After several experiments, the data was partitioned in 60% for the training set, and 20% for both validation and test sets. These sets were originated using a fixed random seed across all experiments, with equal proportions of stage distribution across them.

The stage distribution of the different data used in the experiments is displayed in table 6.1. While the stage distribution is not considerably unbalanced for the data associated to sCr, it is for the data associated with 2B. Thus, the class weights are fed to the loss function.

6.1.3 Hyperparameters and architecture choices

As said earlier, the authors from the Transformer did not specify why they decided to use 6 stacks of attention blocks, so in this study several N number of attention blocks were tested, and the final choice ended up being to work with $N = 10$. Regarding other network hyperparameters, the dense interpolation factor was manually set to 4 after testing the results with different values. The number of heads in the multi-head attention was set to 8, just like in the original architecture, since the results achieved in the experiments with other number of heads in the attention module didn't produce significant changes. Just like in the original architecture, the Adam optimizer with parameters β_1 = 0.9, β_2 = 0.98 was used, with a learning rate of 0.00025. Regarding the attention and residue dropout regularizations, the dropout

probability was set to 0.2, and the batch and embedding sizes were set to 256. Also, the number of epochs during training was set to 100 for all experiments in this section.

6.2 Experiments

6.2.1 Using the sCr classification system

6.2.1.1 All features

The performance of the model was similar across the different sequence lengths used. The best results were achieved when working with 24h sequences, achieving higher scores in pretty much every metric used, as can be seen in Table 6.2. The overall accuracy was 98.2%, with F1 scores of 0.981, 0.975, 0.985 and 0.986 for stages 0 to 3, respectively. Only the F1 score for AKI stage 0 was higher when using a sequence length of 12h, achieving a score of 0.982. The AUC scores for the 24h sequences were 0.993, 0.978, 0.983 and 0.995 for stages 0 to 3, respectively, shown in Figure 6.1 along with the ROC curves.

sCr classification system

Sequence length	AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
6	0	0.969	0.980	**0.992**	0.954	0.973
	1		**0.990**	0.947	0.972	0.959
	2		**0.995**	0.959	0.985	0.971
	3		0.993	0.982	0.973	0.977
12	0	0.975	0.989	0.991	0.974	**0.982**
	1		**0.990**	0.960	0.972	0.966
	2		**0.995**	0.957	**0.987**	0.972
	3		0.992	**0.995**	0.966	0.980
24	0	**0.982**	**0.996**	0.972	**0.990**	0.981
	1		**0.990**	**0.976**	**0.973**	**0.975**
	2		0.994	**0.986**	0.985	**0.985**
	3		**0.994**	**0.995**	**0.977**	**0.986**

Table 6.2: Results of the model using the sCr classification system

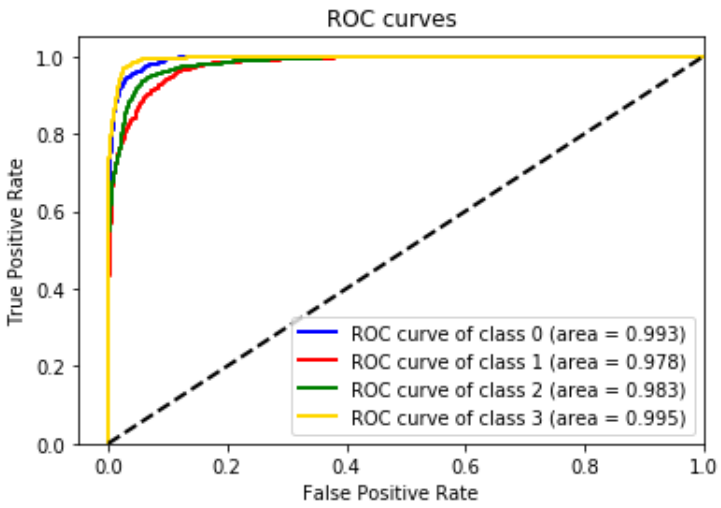


Figure 6.1: ROC curves using 24h sequences

6.2.1.2 10 most important features

The 10 most important features for the sCr classification systems obtained in section 5.4 (shown in Appendix C) are used in these experiments, and shown in Table 6.3. Contrarily to when using all features, the performance of the model this time ended up being slightly better when using sequences of 6h and 12h (shown in Table 6.4). The overall accuracy was 96.3% on both, and the best F1 scores were 0.972, 0.957, 0.965 and 0.974, for stages 0 to 3 respectively. The scores for stages 0 and 1 were achieved using sequence lengths of 6h, and stage 2 and 3 achieved using sequence lengths of 12h. The AUC scores for the 24h sequences were 0.997, 0.961, 0.944 and 0.996 for stages 0 to 3, shown in Figure 6.2 along with the ROC curves. Despite the better overall performance of the model when using all features, the AUC scores were similar comparing to the scores when using only 10 features.

Feature	Score
AKI Stage	0.334
Creatinine - Baseline value	0.103
Creatinine - Baseline value: Lowest 7 days	0.061
Creatinine - Baseline value: Lowest 48hr	0.051
Weight	0.036
Serum Creatinine	0.035
Creatine Phosphokinase (CPK)	0.025
Red Blood Cell Distribution Width (RDW)	0.020
Blood urea nitrogen	0.018
Alkaline phosphatase	0.017

Table 6.3: Feature importance scores for the sCr classification system

sCr classification system						
Sequence length	AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
6	0	**0.963**	0.993	0.962	0.983	**0.972**
	1		**0.986**	**0.954**	**0.960**	**0.957**
	2		**0.989**	0.949	**0.969**	0.958
	3		0.982	0.997	0.931	0.963
12	0	**0.963**	0.988	**0.964**	0.970	0.967
	1		0.985	0.939	0.956	0.947
	2		0.987	**0.968**	0.962	**0.965**
	3		**0.991**	0.988	**0.963**	**0.974**
24	0	0.944	**0.998**	0.907	**0.993**	0.948
	1		0.969	0.935	0.915	0.925
	2		0.980	0.952	0.946	0.949
	3		0.980	**1**	0.921	0.959

Table 6.4: Results of the model using the sCr classification system and the 10 most important features

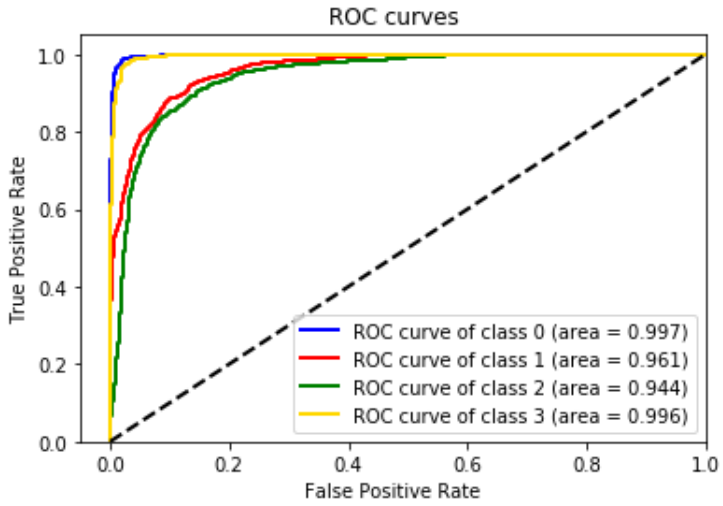


Figure 6.2: ROC curves using 24h sequences and 10 features

6.2.1.3 Results when predicting a stage alteration

The scores achieved were high, although including cases where the last value of the AKI stage in the sequence is the same as the target AKI stage. Using sequence lengths of 24 hours, when looking at the samples whose last AKI stage of the sequence is different from the target, we found out that only 163 from the 26484 total sequences have target values different from the last AKI stage value of the sequences. After the data splitting, only 29 out of the 5297 samples from the test set have different targets and last values of AKI stage in the sequences.

Despite the low proportion of samples with stage alteration, the model using all features is capable of predicting the exact stage in nearly 51.72% of predictions. The accuracy of predicting the exact AKI stage when the patient's diagnosis gets worse was 57.89%, which was the same accuracy for the model to predict if the patient's diagnosis was going to aggravate (correctly predicting the aggravation of the patients AKI condition, instead of simply looking at the correct AKI stage). The model can also predict

the occurrence of episodes of AKI (for patients that were not diagnosed with the disease) with an accuracy of 50%. Table 6.5 displays those predictions, showing the last AKI value of the sequence, the target AKI value and the output produced by the model.

The model performance when using 10 features achieved 41.38% accuracy when predicting the exact stage, 57.89% accuracy predicting the exact AKI stage when the patient's condition worsens, and 63.16% accuracy predicting the aggravation of the patient's AKI condition. The accuracy predicting episodes of AKI for patients with stage 0 by the time of prediction achieved 66.67%. Those predictions are displayed in Table 6.6.

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	2	1-0-1	2	2-1-2	1
0-1-1	1	1-2-1	4	2-3-3	3
0-2-0	1	1-2-2	5	3-0-3	1
0-2-2	2	1-3-1	1	3-1-3	1
1-0-0	3	2-0-0	1	3-2-3	1

Table 6.5: Predictions regarding stage alteration using sCr with all features. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-1	3	1-2-1	5	2-3-3	3
0-2-0	2	1-2-2	4	3-0-3	1
0-2-1	1	1-3-3	1	3-1-3	1
1-0-1	4	2-0-0	1	3-2-3	1
1-0-2	1	2-1-2	1		

Table 6.6: Predictions regarding stage alteration using sCr with 10 features. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

6.2.2 Using the 2B classification system

After realizing the very limited number of samples to predict the accuracy of the model regarding stage changing using sCr, there was a need to find that number of samples in the datasets used in the experiments associated with 2B. As seen in Table 6.1, when working with sequence lengths of 24 hours there are 17078 samples. This time, the number of samples whose last AKI stage of the sequence is different from the target was 1306, almost 10 times more samples compared with sCr. After the data splitting, 253 out of the 3416 total samples from the test set have a change in the AKI stage.

6.2.2.1 All features

The performance of the model was similar across the different sequence lengths used, but slightly better when using 24h sequences. The best overall accuracy was 87.6%, and was achieved when using 6h and 24h sequences (Table 6.7). The best F1 scores were divided across the experiments, with values of 0.931, 0.609, 0.893 and 0.936 for stages 0 to 3, respectively. The best scores for stage 0 was achieved using

sequence lengths of 12h, stage 1 and 3 using sequence lengths of 24h, and stage 2 using sequence lengths of 6h. The AUC scores for the 24h sequences were 0.920, 0.860, 0.920 and 0.990 for stages 0 to 3, respectively, shown in Figure 6.3 along with the ROC curves. It's important to highlight the struggle of the model when predicting stage 1 occurrences, which can be explained by the low proportion of AKI stage 1 in the cohorts regarding the 2B classification system, with only nearly 8% of the samples (Table 6.1).

Regarding the performance of predicting stage alterations (Table 6.8), the model accurately predicts the exact AKI stage in 32.70%, this value increases to 42.58% when looking at the accuracy of predicting the exact stages when the AKI stage increases. Also, the accuracy was 48.39% when predicting the aggravation of the patient's AKI condition, and 53.61% predicting episodes of AKI for patients with stage 0 by the time of prediction.

2B classification system

Sequence length	AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
6	0	**0.876**	**0.967**	0.873	**0.961**	0.915
	1		0.930	**0.770**	0.491	0.599
	2		0.963	0.876	**0.909**	**0.893**
	3		0.982	**0.958**	0.899	0.928
12	0	0.859	0.954	**0.912**	0.951	**0.931**
	1		0.918	0.767	0.448	0.566
	2		**0.965**	0.761	0.894	0.822
	3		0.980	0.918	0.885	0.901
24	0	**0.876**	0.956	0.868	0.951	0.908
	1		**0.952**	0.679	**0.552**	**0.609**
	2		0.926	**0.929**	0.835	0.880
	3		**0.994**	0.912	**0.962**	**0.936**

Table 6.7: Results of the model using the 2B classification system

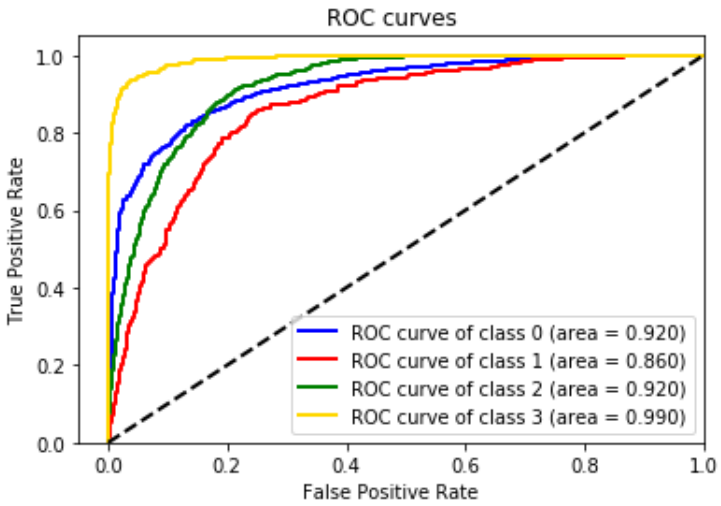


Figure 6.3: ROC curves using 24h sequences

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	40	1-0-2	1	2-1-0	1
0-1-1	34	1-2-0	1	2-1-1	3
0-1-2	6	1-2-1	22	2-1-2	4
0-2-0	5	1-2-2	18	2-3-2	12
0-2-1	3	2-0-0	5	2-3-3	5
0-2-2	9	2-0-1	2	3-0-0	1
1-0-0	4	2-0-2	58	3-2-2	7
1-0-1	16	2-0-3	2	3-2-3	4

Table 6.8: Predictions regarding stage alteration using 2B with all features. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

6.2.2.2 10 most important features

The 10 most important features for the 2B classification systems obtained in section 5.4 used in these experiments are shown in Table 6.9. The performance of the model here was once again better when using 24h sequences (Table 6.10). It produced overall accuracy of 89.2%, and better F1 scores for stages 1, 2 and 3 with only stage 0 having a better score while using 12h sequences. The scores were 0.935, 0.664, 0.890 and 0.947 respectively for stages 0 to 3, and the performance was generally better comparing with the experiments working with all features. The AUC scores were 0.945, 0.741, 0.803 and 0.992 for stages 0 to 3 (Figure 6.4), meaning the AUC scores for stages 1 and 2 were worse working with only 10 features.

Regarding the performance of predicting stage alterations (Table 6.11), the model accurately predicts the exact AKI stage in 17.49%, this value increases to 28.39% when looking at the accuracy of predicting the exact stages when the AKI stage increases. Also, the accuracy was 30.97% when predicting the aggravation of the patient's AKI condition and only 28.87% predicting episodes of AKI for patients with stage 0 by the time of prediction.

Feature	Score
AKI Stage	0.265
Urine Output Rate: 12hr	0.167
Urine Output Rate: 6hr	0.157
Urine Output Rate: 24hr	0.135
Heart Rate	0.011
Systolic Blood Pressure	0.011
Mean Blood Pressure	0.010
Diastolic Blood Pressure	0.010
Respiratory Rate	0.009
Temperature	0.009

Table 6.9: Feature importance scores for the 2B classification system

2B classification system

Sequence length	AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
6	0	0.862	**0.975**	0.844	**0.969**	0.902
	1		0.924	0.790	0.477	0.595
	2		0.963	0.854	0.908	0.880
	3		0.966	**0.980**	0.822	0.894
12	0	0.889	0.970	**0.904**	0.968	**0.935**
	1		0.932	**0.798**	0.503	0.617
	2		**0.967**	0.861	**0.909**	0.885
	3		0.990	0.945	0.941	0.943
24	0	**0.892**	0.950	0.891	0.945	0.918
	1		**0.968**	0.679	**0.650**	**0.664**
	2		0.933	**0.935**	0.849	**0.890**
	3		**0.994**	0.932	**0.962**	**0.947**

Table 6.10: Results of the model using the 2B classification system and the 10 most important features

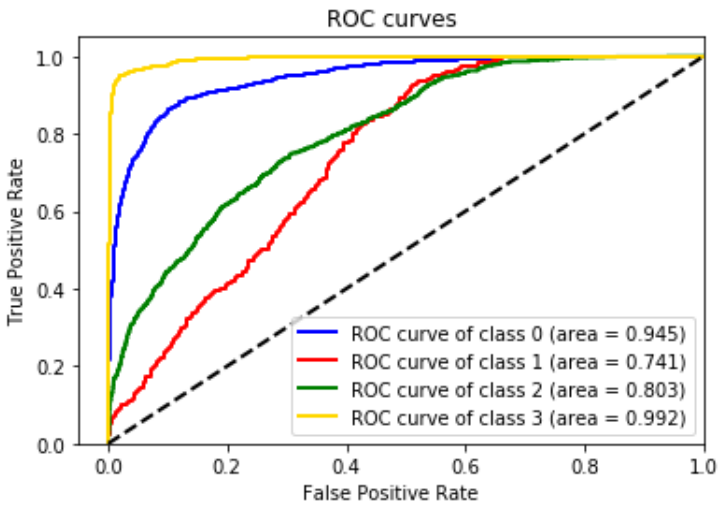


Figure 6.4: ROC curves using 24h sequences and 10 features

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	57	1-0-2	5	2-1-2	7
0-1-1	22	1-2-1	22	2-3-2	16
0-1-2	1	1-2-2	19	2-3-3	1
0-2-0	12	2-0-0	1	3-0-3	1
0-2-1	3	2-0-2	65	3-2-2	2
0-2-2	2	2-0-3	1	3-2-3	9
1-0-1	16	2-1-1	1		

Table 6.11: Predictions regarding stage alteration using 2B with 10 features. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

6.2.2.3 Using different learning rates

While the experiments using the sCr classification system were consistent in terms of results. This didn't happen when working with the 2B classification system, as the learning rate value had implications not only on the overall performance but also on the accuracy of stage alteration occurrences. The experiments before showed that, in general, the model produced better results when using 24h sequences. Thus, the experiments in this section only work with 24h sequences.

Comparing both experiments, the reduced number of features produced better overall accuracy and better F1 scores for stages 0 and 1, while the model using all features produced better scores for stage 2 and 3.

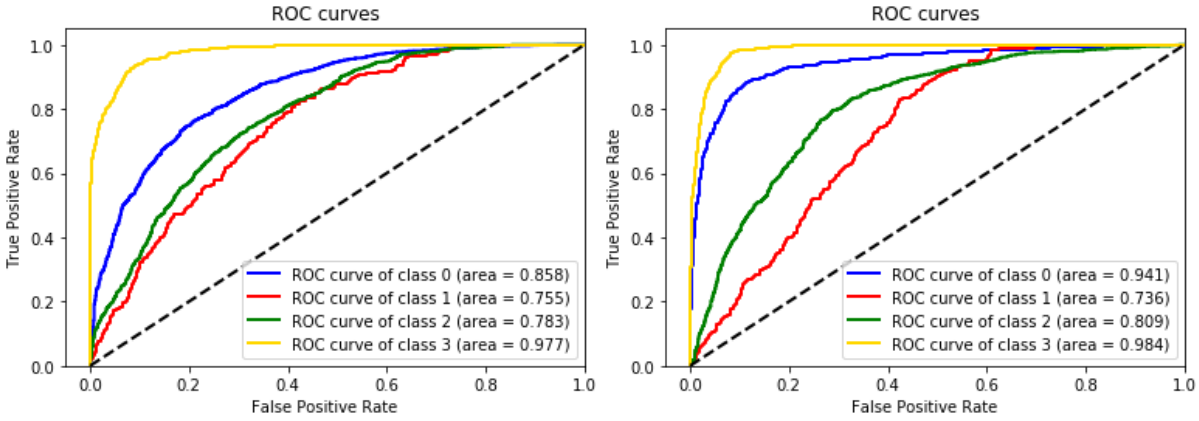
Regarding the model performance of predicting stage alterations in these experiments (Tables 6.13 and 6.14), the model achieved better performance using only 10 features, as it achieved 32.50% accuracy predicting the exact AKI stage, 60.65% accuracy predicting the exact stages when the AKI stage increases, and 63.26% when predicting the aggravation of the patient's AKI condition. Using all features, those values dropped down to 32.32%, 45.16% and 54.84%, respectively. Contrarily to that, the performance of the model predicting AKI episodes for patients without the condition at the time of prediction was better using all features: 64.95% versus the 56.70% achieved by the model using only 10 features.

As addressed before, the performance using a lower learning rate resulted in a worst general performance of the model, but a much higher accuracy predicting the stage alteration of the patient.

2B classification system

	AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
All features	0	0.795	0.958	0.765	0.947	0.846
	1		0.899	0.642	0.357	0.459
	2		0.919	0.811	0.801	0.806
	3		0.962	0.957	0.799	0.871
10 features	0	0.814	0.966	0.801	0.959	0.873
	1		0.940	0.606	0.470	0.530
	2		0.910	0.808	0.785	0.796
	3		0.941	0.991	0.727	0.839

Table 6.12: Results of the model using the 2B classification system using a learning rate value of 0.0001



(a) All features (b) 10 features

Figure 6.5: ROC curves

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	30	1-0-1	14	2-1-0	2
0-1-1	43	1-0-2	2	2-1-2	6
0-1-2	6	1-2-0	2	2-3-2	6
0-1-3	1	1-2-1	28	2-3-3	11
0-2-0	4	1-2-2	11	3-0-3	1
0-2-1	7	2-0-0	9	3-2-2	1
0-2-2	5	2-0-1	6	3-2-3	10
0-2-3	1	2-0-2	45		
1-0-0	5	2-0-3	7		

Table 6.13: Predictions regarding stage alteration using 2B with all features and a learning rate of 0.0001. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	34	1-0-2	12	2-1-2	7
0-1-1	44	1-2-1	12	2-3-2	3
0-1-2	2	1-2-2	28	2-3-3	14
0-2-0	8	1-2-3	1	3-0-3	1
0-2-1	1	2-0-2	59	3-2-2	1
0-2-2	8	2-0-3	8	3-2-3	10
1-0-1	9	2-1-1	1		

Table 6.14: Predictions regarding stage alteration using 2B with 10 features and a learning rate of 0.0001. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

6.3 Experiments with more focus on predicting stage alteration

Despite the really low proportion of examples where the AKI stage changes from the last value of the sequence into the target value, the model is capable of accurately predicting when it happens, and even if the exact stage is not correct, the model can still predict when the patient's condition gets worse.

In these following experiments, the training samples used will be more representative of alterations regarding the patient's AKI stage. This task is focused on understanding the model's capacity when working with a much more balanced dataset in that regard. Also, the experiments in this section were tested with 10 and all features, but since the model produced better results using all features only those experiments will be displayed.

For the dataset associated with sCr, since only 163 data samples were cases where the AKI stage did change, the dataset used consisted in those 163 samples mixed with 50 samples from each stage when the AKI stage remained the same. This means the dataset used will have 363 samples, with pretty much half of them being samples with stage alteration. Regarding 2B, since 1306 samples had stage alteration, those samples were mixed with 500 examples for each stage that did not change stage, ending up with a dataset of 3306 samples.

After splitting the datasets with the same methodology as in the first experiments, the test sets had stage alteration samples of 34 out of 73, and 250 out of 662, for sCr and 2B respectively. The batch size was decreased following the reduction of samples for both datasets regarding each classification system. For sCr the batch size selected was 32, and 64 for 2B.

6.3.1 sCr classification system

The overall performance in these experiments were worse, as expected. The model achieved an overall accuracy of 52.1%, and F1 scores of 0.545, 0.400, 0.468 and 0.667 for stages 0 to 3 respectively (Table 6.15). The sensitivity was really low for stage 1, with a score of 0.286, contrarily to stage 3 as it reached a score of 0.800. The performance of the model was generally better for stage 3 predictions, also supported by the AUC scores in Figure 6.6. The AUC score for stage 3 was 0.830, clearly higher compared to the other stages: 0.695, 0.727 and 0.607 respectively for stages 0 to 2.

Regarding the performance of predicting stage alterations (Table 6.16), the model accurately predicts the exact AKI stage in 29.41%, this value increases to 31.81% when looking at the accuracy of predicting the exact stages when the AKI stage increases. Also, the accuracy of predicting the aggravation of the patient's AKI condition was 59.09%, and 66.67% when predicting AKI episodes for patients without the condition.

sCr classification system

AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
0	0.521	0.860	0.563	0.529	0.545
1		0.942	0.286	0.667	0.400
2		0.712	0.524	0.423	0.468
3		0.845	0.8	0.571	0.667

Table 6.15: Results of the model using 24h sequences and the sCr classification system

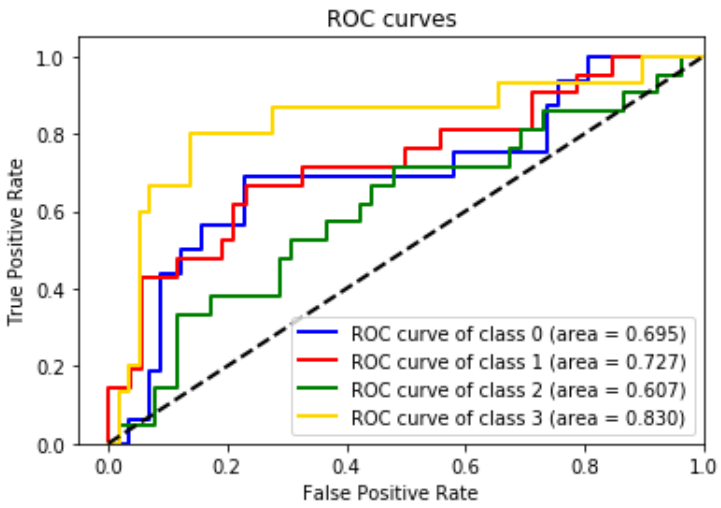


Figure 6.6: ROC curves 24h sequences and sCr

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	2	1-0-2	2	2-3-0	1
0-1-1	1	1-2-0	1	2-3-1	1
0-1-2	4	1-2-1	1	2-3-3	1
0-1-3	1	1-2-2	1	3-0-0	1
0-2-0	2	1-3-0	2	3-0-3	1
0-2-2	1	1-3-3	1	3-1-0	1
0-2-3	1	2-0-2	2		
1-0-0	2	2-1-2	4		

Table 6.16: Predictions regarding stage alteration using 24h sequences and sCr. LS - last AKI stage value in the sequence; T - target value, P - prediction output

6.3.2 2B classification system

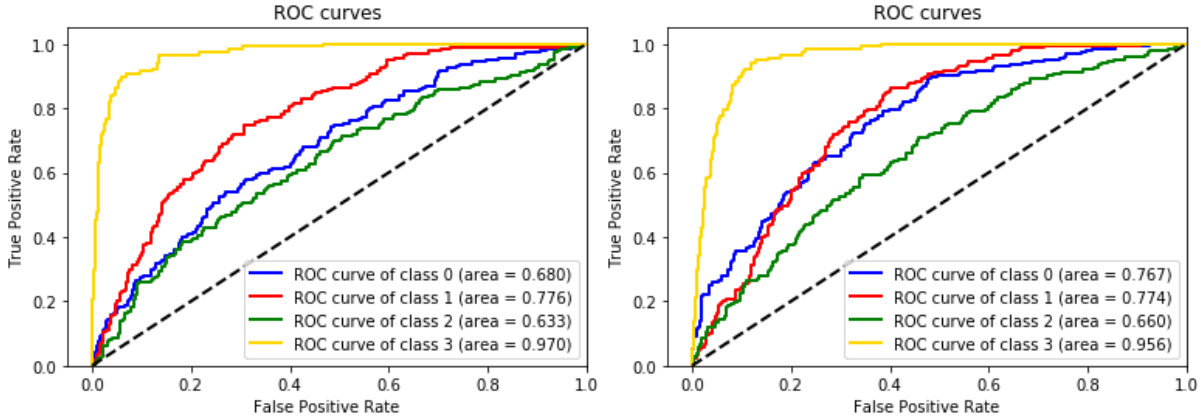
Once again different learning rates were tested, and the best overall performance was achieved using the lowest value (Table 6.17). It managed to produce an overall accuracy of 55.9%, and the higher F1 scores for stages 0, 1 and 3 (0.521, 0.521 and 0.792) while the higher learning rate achieved higher F1 score for stage 2 (0.502). The performance of the model was once again better for stage 3 predictions, also supported by the AUC scores in Figure 6.7. The AUC score for stage 3 was over 0.950 for both experiments, clearly higher compared to all the other stages which have scores below 0.800.

Contrarily to the overall performance of the model, the performance regarding stage alteration was better using the higher learning rate (Tables 6.18 and 6.19). The model achieved 33.20% accuracy predicting the exact stage, 44.12% accuracy predicting the exact stage when the AKI condition worsens, and 58.82% of accuracy predicting the aggravation of the patient's AKI condition. The lower learning rate achieved 31.60%, 36.76% and 44.85%, respectively, for the metrics addressed before. The performance predicting AKI episodes for patients without the condition was also better using the higher learning rate, achieving 68.05% accuracy, higher than the 62.50% produced by the model using the lower learning rate.

2B classification system

Learning Rate	AKI Stage	Overall Accuracy	Specificity	Sensitivity	Precision	F1 score
0.00025	0	0.559	0.864	0.476	0.575	0.521
	1		0.810	0.534	0.508	0.521
	2		0.802	0.462	0.469	0.465
	3		0.927	0.880	0.720	0.792
0.0005	0	0.544	0.908	0.373	0.611	0.463
	1		0.853	0.438	0.523	0.477
	2		0.696	0.604	0.430	0.502
	3		0.925	0.880	0.715	0.789

Table 6.17: Results of the model using 24h sequences and the 2B classification system



(a) Learning rate of 0.00025

(b) Learning rate of 0.0005

Figure 6.7: ROC curves using 24h sequences and the 2B classification system

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	20	1-0-2	4	2-0-3	4	3-0-3	1
0-1-1	31	1-0-3	1	2-1-0	2	3-1-2	1
0-1-2	9	1-2-0	6	2-1-1	2	3-2-0	1
0-2-0	7	1-2-1	35	2-1-2	5	3-2-1	2
0-2-1	2	1-2-2	10	2-1-3	1	3-2-2	4
0-2-2	3	2-0-0	19	2-3-2	7	3-2-3	8
1-0-0	3	2-0-1	9	2-3-3	6		
1-0-1	17	2-0-2	29	3-0-0	1		

Table 6.18: Predictions regarding stage alteration using 2B and a learning rate of 0.00025. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

LS-T-P	Count	LS-T-P	Count	LS-T-P	Count	LS-T-P	Count
0-1-0	19	1-0-1	13	2-0-1	1	2-3-3	9
0-1-1	23	1-0-2	8	2-0-2	38	3-0-2	1
0-1-2	18	1-0-3	2	2-0-3	8	3-0-3	1
0-2-0	4	1-2-0	2	2-1-0	1	3-1-2	1
0-2-1	2	1-2-1	27	2-1-1	3	3-2-1	1
0-2-2	6	1-2-2	22	2-1-2	6	3-2-2	4
1-0-0	2	2-0-0	14	2-3-2	4	3-2-3	10

Table 6.19: Predictions regarding stage alteration using 2B and a learning rate of 0.0005. (LS - last AKI stage value in the sequence; T - target value, P - prediction output)

6.4 Chapter discussion

In this chapter, several experiments regarding the model were made. Starting with the architecture, different parameters were tested, including the original from Song et. al [3], and the architecture that achieved better results was used across all experiments in this chapter. The performance of the model was tested with different sequences lengths, and in general, the best results were achieved using longer sequences.

The number of features used in the experiments seemed to affect the performance of the model. Using the sCr classification system, the model produced better overall results when using all features, but got the best results regarding stage alteration when using only 10 features. When using the 2B classification system, using 10 features achieved better results overall and regarding stage alterations, comparing to the experiments with all features. Also, although experimenting with distinct learning rate values did not show major differences when using the sCr classification system, it did influence the experiments using 2B.

The overall performance of the model was better using the sCr classification system, which comes as no surprise knowing the low number of samples whose stage actually changes. The data associated with 2B has a higher proportion of samples with stage alterations and achieved lower scores in the metrics, showing that the model performs well predicting the continuity of the same stage.

The specificity values were high across all experiments in this section, which comes as no surprise knowing that there are a lot more true negatives due to merging the samples for 3 stages.

In the end, despite experimenting with reduced samples the best performance achieved by the model for both classification systems was through working with the cohort from the initial experiments. Looking at the metrics used across the experiments, and knowing that the proportions of samples with stage alterations are lower, it's clear that the model is able to correctly predict when the patient's AKI stage remains the same. Regarding the stage alteration performance, the best results achieved up to 63.16% accuracy predicting the aggravation of the patient's AKI condition using the sCr classification system, and 63.26% using 2B. Also, the model was able to achieve 32.50% accuracy predicting the exact AKI stage, 60.65% accuracy predicting the exact stages when the AKI stage increases using 2B, as well as 41.38% and 57.89% for those same metrics using the sCr classification system.

The model had difficulties predicting the exact AKI stage across all experiments, achieving 33.30% of accuracy as its best score using 2B, and 51.72% using sCr. When predicting the occurrence of an AKI episode for patients with stage 0 at time of prediction, the best results achieved were 68.05% accuracy using the 2B classification system, and 66.67% using sCr. While the best performance for sCr was exactly the same in the initial experiments and the experiments with reduced data, the model using 2B achieved better scores with a much more balanced dataset regarding samples with stage alteration and stages that stayed the same. All the confusion matrices for the experiments in this chapter can be seen in Appendix F

Chapter 7

Conclusions

This thesis set out to study the progression of AKI on ICU patients using a self-attention model. While studying the progression of the disease, two distinct variations of the KDIGO classification system were tested. One only focusing on serum creatinine values of the patient to define its AKI stage, which was labelled as sCr, while the other used the criteria regarding both serum creatinine and urine output. The latter was labelled as 2B, and the procedure associated with it in this work was, to the best of our knowledge, the first time it was used.

Along the development of this work, some decisions were made with the thought process of replicating how a caretaker would look at the problem. One example of that is the choice of including features that were drawn by the KDIGO criteria for both serum creatinine and urine output, as the caretaker can realistically have access to that information and decide to use it. Another example is the methods used during the missing data imputation segment, where last observation carried forward and next observation carried backward were chosen over other interpolation methods.

The self-attention model used in this work was tested using different parameters when comparing to the original architecture. The original architecture uses $N = 6$ stacks of attention blocks, and since the performance of the model was better using 10 all the experiments were made using $N = 10$. While the number of heads was kept at 8 just like the original architecture, the dense interpolation factor was manually set to 4 after testing the results with different values.

Besides the details about the architecture, there were other interesting topics to address, such as the performance of the model when using different sequence lengths and also if the number of features affected the results. While the overall performance and accuracy predicting stage alterations was better working with the longer sequences, the different number of features seems to influence the performance in some predictions. The best accuracy predicting the occurrence of an AKI episode was achieved using all features for both stages, and predicting the aggravation of the patient's AKI condition was achieved using only 10 features.

Comparing the best results for each classification system, 2B achieved better results regarding the predictions of stage alterations. The accuracy predicting the occurrence of an AKI episode for patients without AKI at time of prediction reached 68.05% using 2B and 66.67% using sCr. Both of those scores were achieved during the experiments with reduced samples, which used all features. This showed that the model has higher performance predicting the occurrence of an AKI episode when there's more balance between samples whose stage change and samples that maintain the AKI stage, for both classification systems. The model's best performance predicting the aggravation of the patient's AKI condition achieved 63.26% and 63.16% for 2B and sCr, respectively, both using 10 features. The performance using 2B might be better than sCr due to the extremely low number of samples with stage alterations, meaning

there are more training examples with the alterations possible such as going from stage 0 to stage 2, or the other way around.

The work from Tomašev et al. [42] studied the continuous prediction of AKI using RNNs, with the goal of predicting the occurrence of AKI episodes within 48 hours of the time of prediction. Using an extensive cohort of 703,782 patients, with information regarding the patients prior to their ICU stay, the authors used the KDIGO classification system only focusing on the values of serum creatinine to determine the AKI stages. The model was able to predict 55.80% of all episodes of AKI for patients in the ICU, with a lead time of up to 48 hours.

Although not comparing directly the results from both studies, since this study focused on predicting the patient's AKI stage in the following hour, the results achieved in this work indicate that the self-attention model joint with the methods used ends up having a higher performance when predicting of the occurrence of AKI in patients that were not diagnosed with the disease at the time of prediction. Also, the model's capability to correctly predict the aggravation of a patient's AKI condition is interesting and indicates that not only the model may be able to replicate the results when focusing on predicting the occurrence of AKI within the following 48 hours, but also predict the aggravation of the AKI condition within several hours in advance.

In the paper from the self-attention model used in this work, Song et. al [3] proved that using their self-attention model outperformed state-of-the-art RNNs. Since the work from Tomašev et al. used a RNN architecture, along with the fact that 2B achieved higher performance than sCr, which was used in their work, also gives an idea that the methods and self-attention model used in this work can outperform those results.

7.1 Limitations

One of the limitations of this study is in terms of the database. MIMIC-III is a very large database, and as it was explained in the Data Pre-Processing chapter (Chapter 4) it consists of two different information systems. Each system has their own ID code for their measures, meaning that the same measure can have a lot of different codes, making it particularly hard to have the entire time series of all the variables. Some of the ID codes had unclear labels, making it very difficult to determine what measures were collected under those ID codes. A noticeable example is Urine Output (UO), as it was a main variable in this work. With over a hundred ID codes associated with it, only the ID codes with more data were used. By not using all the codes, some measures may be missing from the time series making the time difference between measures appear longer than they in fact were. Since this measure uses the time difference, it is possible that it was assigned a value of urine volume per hour smaller than the real value, which can influence the AKI stage. Even though two data extraction resources were used, there was still a need to manually deal with the remaining ID codes. This process ended up being extremely lengthy, and due to the unclear label names some measures that could add some value to the model may have been excluded.

The data format necessary to feed the model can be seen as a limitation, as it demands information for every feature. The consequence of this led to the decision of only using features with records for every patients, which limited the number of features available to use.

Still regarding the database, after the full pre-processing task, the patient cohort kept did not have balanced classes. In regards to the task of working with age groups, it wasn't possible to balance the target classes, and because some classes had extremely low proportions we did not proceed to study the performance of the model across different age groups.

In the experiments, despite the low number of samples where the last AKI value of the sequence fed to the model changed compared to the AKI stage in the target, the model did achieve good results regarding those stage alterations. More samples with stage alterations could mean higher performance for the model. It would be interesting to make a deeper analysis, evaluating the model performance by changing from one AKI stage to another (for example stage 0 to stage 3, and vice versa) but that would require more data.

7.2 Future Work

As addressed in the related work section, both Pereira et al. [44] and Tomašev et al. [42] used confidence level prediction for each of the AKI prediction made. This is an important asset, as it means that the prognostic predictions will have a given uncertainty level associated, solving the trustworthiness issue around the prognostic prediction, that exists in a lot of prognostic models, thus it is appropriate and valuable to use in studies regarding clinical issues.

The same study could be replicated using different baseline values for the SCr segment of the KDIGO classification system, as the choice for this work ended up following the baselines from the works of Silva et al., Correia et al. and Cunha et al. [18, 17, 16], that used the lowest value of the last three measures of SCr. The other two different baselines addressed in the Baseline Estimations section (Section 2.1.2) were the baseline value equal to the sCr value at admission [38] and also the baseline being equal to the lowest value of SCr during the whole stay [39].

Also talked about in the related work section, Pires et al's study [43], the patients were stratified within their own disease progression rate. In the AKI context, this approach idea could be done when using the SCr baseline as the lowest value of the past 7 days, and taking into account the two KDIGO criteria that involve SCr levels alteration through time. The patients positive predicted of having AKI through the 48h criteria being labeled as 'Fast progressors', while the patients indicated with the disease through the 7 day criteria being labeled as 'Slow progressors'. Both sets of patients identified with their AKI stage, and later split into groups of patients with 'AKI stage 1' and 'AKI stage 2 & 3' [42], where the progression of the disease could be studied for every 48h period. With this approach, the goal is to compare results between slow and fast progressors, and also to evaluate the progression within the AKI stages.

In the missing data imputation section (section 4.2.5), the usage of interpolation methods could be used with the goal of best using the model's ability to value the variability within features in the training sequences. Testing the results using Linear or Spline interpolation would be interesting, because while Linear is the simplest method of interpolation, Spline (particularly Cubic Spline Interpolation) is a flexible alternative to polynomial interpolation, reducing the order of the polynomials used, as it fits several smaller-degree polynomials instead of only one complex polinomyal, making it simpler and closer to reality [86].

Some other topics that could be taken into account in future work include the usage of patients data from outside the ICU. Using information prior the patient's entry in the ICU might be interesting. Also, using data from other tables in MIMIC might be valuable, particularly information regarding medication administered to the patient, on the INPUTEVENTS table.

As already addressed, in this thesis only the prognostic of the following hour is considered. With positive results in this procedure, the next step is to study the same methodology but predicting for several hours ahead, such as trying to predict the AKI stage of a patient 24h later, when training the model with

sequences whose length could also be larger than the 24h used in this work. Replicate the approach by Tomašev et al. [42] would be interesting, in terms of predicting up to 24h, 48h or 72h ahead the occurrence of AKI, where a comparison between the results produced by the RNN used in their work and the self-attention architecture used in this could be analyzed, knowing that SAnD produced better results on the clinical tasks for the developers of the model, compared to RNNs. One problem with this approach could be the limitation of the data provided by MIMIC-III, as the cohort used by Tomašev et al. consisted in 703,782 patients, against the 44,476 patients provided by MIMIC-III (before the exclusion criteria).

References

[1] Pranay Dugar. Attention — seq2seq models. https://towardsdatascience.com/day-1-2-attention-seq2seq-models-65df3f49e263, Jul 2019. Medium. XIII, 14, 15, 16

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. XIII, 14, 16, 18, 21, 22, 25

[3] Huan Song, Deepta Rajan, Jayaraman J. Thiagarajan, and Andreas Spanias. Attend and diagnose: Clinical time series analysis using attention models, 2017. XIII, 2, 21, 22, 64, 66

[4] Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):160, Mar 2021. 1

[5] Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *2011 International Conference on Electronics, Communications and Control (ICECC)*, pages 1028–1031, 2011. 1

[6] Sam Musa. Smart cities - a roadmap for development. *Journal of Telecommunications System Management*, 05, 01 2016. 1

[7] Benefits of EHRs. https://www.healthit.gov/topic/health-it-and-health-information-exchange-basics/benefits-ehrs. The Office of the National Coordinator for Health Information Technology (ONC). 1

[8] Sergio Sánchez Martínez, Oscar Camara, Gemma Piella, Maja Cikes, Miguel Ángel González Ballester, Marius Miron, Alfredo Vellido, Emilia Gómez, Alan Fraser, and Bart Bijnens. Machine learning for clinical decision-making: Challenges and opportunities, 11 2019. 2

[9] D. H. Li, R. Wald, D. Blum, E. McArthur, M. T. James, K. E. A. Burns, J. O. Friedrich, N. K. J. Adhikari, D. M. Nash, G. Lebovic, A. K. Harvey, S. N. Dixon, S. A. Silver, S. M. Bagshaw, and W. Beaubien-Souligny. Predicting mortality among critically ill patients with acute kidney injury treated with renal replacement therapy: Development and validation of new prediction models. *J Crit Care*, 56:113–119, 04 2020. 2

[10] P. Radha and R. Divya. Multiple time series clinical data with frequency measurement and feature selection. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pages 250–254, 2016. 2

[11] Siteng Huang, Donglin Wang, Xuehan Wu, and Ao Tang. Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, November 2019. 2

[12] Kidney Disease Improving Global Outcomes (KDIGO) Acute Kidney Injury Work Group. KDIGO Clinical Practice Guideline for Acute Kidney Injury. *Kidney International Supplements*, 2:1–138, 2012. 2, 3, 5

[13] Henry Wang, Paul Muntner, Glenn Chertow, and David Warnock. Acute kidney injury and mortality in hospitalized patients. *American journal of nephrology*, 35(4):349–355, 2012. 2, 8

[14] James F. Doyle and Lui G. Forni. Acute kidney injury: short-term and long-term effects. *Critical Care*, 20(1):188, Jul 2016. 3

[15] Danielle Saly, Alina Yang, Corey Triebwasser, Janice Oh, Qisi Sun, Jeffrey Testani, Chirag R. Parikh, Joshua Bia, Aditya Biswas, Chess Stetson, Kris Chaisanguanthum, and F. Perry Wilson. Approaches to predicting outcomes in patients with acute kidney injury. *PLOS ONE*, 12(1):1–12, 01 2017. 3

[16] Vanessa S. Cunha, Cátia M. Salgado, Susana M. Vieira, and João M. C. Sousa. Fuzzy modeling to predict short and long-term mortality among patients with acute kidney injury. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 148–153, 2016. 3, 8, 32, 67

[17] Clara C. Correia. Data-based modelling for the prediction of mortality in acute kidney injury patients. Master's thesis, Instituto Superior Técnico, July 2017. 3, 8, 28, 29, 32, 67

[18] A. D. Silva, C. C. Correia, C. M. Salgado, S. Finkelstein, L. M. Celi, J. M. C. Sousa, and S. M. Vieira. Fuzzy modeling for predicting patient survival rate in icu with aki. *Institute of Electrical and Electronics Engineers*, pages 1–6, 2018. 3, 8, 32, 67

[19] Rinaldo Bellomo, Claudio Ronco, John Kellum, Ravindra Mehta, and Paul Palevsky. Acute renal failure - definition, outcome measures, animal models, fluid therapy and information technology needs: the second international consensus conference of the acute dialysis quality initiative (adqi) group. *Critical Care*, 8(4):(R204), 2004. 5, 6

[20] Dinna Cruz, Zaccaria Ricci, and Claudio Ronco. Clinical review: Rifle and akin – time for reappraisal. *Critical Care*, 13:(211), 2009. 5

[21] David Warnock. Towards a definition and classification of acute kidney injury. *Journal of the American Society of Nephrology*, 16(11):3149–3150, 2005. 5

[22] J. A. Kellum. Why are patients still getting and dying from acute kidney injury. *Current opinion in critical care*, 22:513–519, 2016. 5

[23] N. H. Lameire et al. Acute kidney injury: an increasing global concern. *Lancet*, 382:170–179, 2013. 5

[24] R. L. Mehta et al. International society of nephrology's 0by25 initiative for acute kidney injury (zero preventable deaths by 2025): a human rights case for nephrology. *Lancet*, 385:2616–2643, 2015. 5

[25] E. A. J. Hoste, J. A. Kellum, N. M. Selby, et al. Global epidemiology and outcomes of acute kidney injury. *Nature Reviews Nephrology*, 14:607–625, 2018. 5

[26] Melanie Meersch et al. Long-term clinical outcomes after early initiation of rrt in critically ill patients with aki. *Journal of the American Society of Nephrology : JASN*, 29(3):1011–1019, 2018. 5

[27] A. Bihorac et al. Long-term risk of mortality and acute kidney injury during hospitalization after major surgery. *Annals of Surgery*, 249:851–858, 2009. 5

[28] J. A. Kellum, F. E. Sileanu, A. Bihorac, E. A. Hoste, and L. S. Chawla. Recovery after acute kidney injury. *American journal of respiratory and critical care medicine*, 195:784–791, 2017. 6

[29] L. S. Chawla, P. W. Eggers, R. A. Star, and P. L. Kimmel. Acute kidney injury and chronic kidney disease as interconnected syndromes. *N. Engl. J. Med*, 371:58–66, 2014. 6

[30] F. Depret, M. Prud'homme, and M Legrand. A role of remote organs effect in acute kidney injury outcome? *Nephron*, 137:273–276, 2017. 6

[31] P. C. Wu et al. Long-term risk of upper gastrointestinal hemorrhage after advanced aki. *Clinical Journal of the American Society of Nephrology*, 10:353–362, 2015. 6

[32] M. T. James et al. Associations between acute kidney injury and cardiovascular and renal outcomes after coronary angiography. *Circulation*, 123:409–416, 2011. 6

[33] C. Arias-Cabrales et al. Short- and long-term outcomes after non-severe acute kidney injury. *Clinical and experimental nephrology*, 22:61–67, 2018. 6

[34] L. S. Chawla et al. Association between aki and long-term renal and cardiovascular outcomes in united states veterans. *Clinical Journal of the American Society of Nephrology*, 9:448–456, 2014. 6

[35] A. O. Kük N. K. İlkaya N. Murat B. Bilgiç H. Abanoz Eur J F. Ülger, M. Pehlivanlar Kük. Evaluation of acute kidney injury (aki) with rifle, akin, ck, and kdigo in critically ill trauma patients. *European journal of trauma and emergency surgery: official publication of the European Trauma Society*, 2017. 6, 9

[36] Namyong Park, Eunjeong Kang, Minsu Park, Hajeong Lee, Hee-Gyung Kang, Hyung-Jin Yoon, and U. Kang. Predicting acute kidney injury in cancer patients using heterogeneous and irregular data. *PLOS ONE*, 13(7):1–21, 07 2018. 6

[37] Sushrut Waikar, Rebecca Betensky, and Joseph Bonventre. Creatinine as the gold standard for kidney injury biomarker studies? *Nephrology Dialysis Transplantation*, 24(11):3263–3265, 2009. 7

[38] Sean Bagshaw. Short-and long-term survival after acute kidney injury. *Nephrology Dialysis Transplantation*, 23(7):2126–2128, 2008. 7, 67

[39] Tal Mandelbaum, Daniel Scott, Joon Lee, Roger Mark, Atul Malhotra, Sushrut Waikar, Michael Howell, and Daniel Talmor. Outcome of critically ill patients with acute kidney injury using the akin criteria. *Critical care medicine*, 39(12):2659, 2011. 7, 67

[40] A. Johnson, T. Pollard, L. Shen, et al. Mimic-iii, a freely accessible critical care database. *Sci Data*, 3(160035), 2016. 8, 27

[41] sonia Yaqub, Kashif Kazmi, kashif, Hasanat Sharif, and Shiraz Hashmi. COMPARISON OF DEFINITIONS (RIFLE, AKIN, AND KDIGO) OF ACUTE KIDNEY INJURY FOR PREDICTION OF OUTCOMES IN ADULTS AFTER ISOLATED CORONARY ARTERY BYPASS GRAFT(CABG) SURGERY. *Nephrology Dialysis Transplantation*, 34(Supplement$_1$), $June$2019. 9

[42] N. Tomašev, X. Glorot, J. W. Rae, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572:116–119, 2019. 9, 32, 35, 36, 66, 67, 68

[43] S. Pires, M. Gromicho, S. Pinto, M. Carvalho, and S. C. Madeira. Predicting non-invasive ventilation in als patients using stratified disease progression groups. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 748–757, 11 2018. 12, 35, 67

[44] Telma F. L. M. Pereira. *Prognostic models targeting time to conversion, stable predictors, and reliability at patient-level: Predicting progression from mild cognitive impairment to dementia*. PhD thesis, Instituto Superior Técnico, 2019. 13, 67

[45] Alzheimer's Association. *Changing the Trajectory of Alzheimer's Disease: How a Treatment by 2025 Saves Lives and Dollars*. Tech. Rep., Chicago, Illinois, USA, 2015. 13

[46] J. Cummings, G. Lee, T. Mortsdorf, A. Ritter, and K. Zhong. Alzheimer's disease drug development pipeline: 2017. *Alzheimer's and Dementia: Translational Research and Clinical Interventions*, 3(3):367–384, 2017. 13

[47] Gabriel Loye. Attention mechanism. `https://blog.floydhub.com/attention-mechanism/`, September 2018. Floydhub. 14

[48] A comprehensive guide to attention mechanism in deep learning for everyone. `https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning/`, 2019. Analytics Vidhya. 14

[49] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 1409.0473. 14

[50] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models, 2019. 1906.05909. 14

[51] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. 14

[52] Nagesh Chauhan. Attention mechanism in deep learning, explained. `https://www.kdnuggets.com/2021/01/attention-mechanism-deep-learning-explained.html`, 2020. KDnuggets. 14

[53] Stefania Cristina. The attention mechanism from scratch. `https://machinelearningmastery.com/the-attention-mechanism-from-scratch/`, September 2020. Machine Learning Mastery. 14

[54] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading, 2016. 1601.06733. 16

[55] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. 16

[56] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization, 2017. 1705.04304. 16

[57] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding, 2017. 1703.03130. 16

[58] Jay Alammar. The illustrated transformer. Retrieved from `http://jalammar.github.io/illustrated-transformer/`, 2018. Blog post. 19

[59] Yoon Kim. Convolutional neural networks for sentence classification, 2014. 1408.5882. 22

[60] Andrew Trask, David Gilmore, and Matthew Russell. Modeling order in neural word embeddings at scale, 2015. 1506.02338. 23

[61] Crossentropyloss. Retrieved from `https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html`. PyTorch. 24

[62] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 25

[63] Hirotaka Kawashima. SAnD. `https://https://github.com/khirotaka/SAnD`, 2019. GitHub repository. 25

[64] Comet. `https://www.comet.ml/site/`. 26

[65] A. Johnson, T. Pollard, and R. Mark. *MIMIC-III Clinical Database*. PhysioNet, 2016. 27

[66] Furukawa MF Charles D, Gabriel M. Adoption of electronic health record systems among u.s. non-federal acute care hospitals: 2008-2013, May 2014. ONC Data Brief, no. 16. 27

[67] Francis S. Collins and Lawrence A. Tabak. Policy: Nih plans to enhance reproducibility. *Nature*, 505:612–613, 2014. 27

[68] Leo Anthony Celi, Robin Tang, Mauricio Villarroel, Guido Davidzon, William Lester, and Henry Chueh. A clinical database-driven approach to decision support: Predicting mortality among patients with acute kidney injury. *Journal of Healthcare Engineering*, 2(1):97–110, 2011. 29

[69] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019. 29, 32

[70] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. Mimic-extract. *Proceedings of the ACM Conference on Health, Inference, and Learning*, Apr 2020. 29

[71] The glasgow structured approach to assessment of the glasgow coma scale. `https://www.glasgowcomascale.org/what-is-gcs/`. Royal College of Physicians and Surgeons of Glasgow. 30

[72] Si conversion calculator. https://www.amamanualofstyle.com/page/si-conversion-calculator. Oxford University Press. 31

[73] Matthias Mack and Alexander R. Rosenkranz. Basophils and mast cells in renal injury. *Kidney international*, 76(11):1142–1147, Dec 2009. 19692999[pmid]. 31

[74] Jens Rocktaeschel, Hiroshi Morimatsu, Shigehiko Uchino, Donna Goldsmith, Stephanie Poustie, David Story, Geoffrey Gutteridge, and Rinaldo Bellomo. Acid-base status of critically ill patients with acute renal failure: analysis based on stewart-figge methodology. *Critical care (London, England)*, 7(4):R60–R60, Aug 2003. 31

[75] M. Bedford, P. Stevens, S. Coulton, et al. Development of risk models for the prediction of new or worsening acute kidney injury on or during hospital admission: a cohort and nested study. appendix 2, variable relationships with acute kidney injury. *Southampton (UK): NIHR Journals Library;Feb. (Health Services and Delivery Research, No*, 4:6, 2016. 31

[76] CY. Chen, Y. Zhou, P. Wang, et al. Elevated central venous pressure is associated with increased mortality and acute kidney injury in critically ill patients: a meta-analysis. *Crit Care*, 24:80, 2020. 31

[77] Anna Malkina. Acute kidney injury - kidney and urinary tract disorders, Mar 2020. 31

[78] B. J. Thomson, D. McAreavey, J. M. Neilson, R. J. Winney, and Ewing DJ. Heart rate variability and cardiac arrhythmias in patients with chronic renal failure. *Clin Auton ResJun;*, 1(2):131–3, 1991. 31

[79] E. J. Filippone, W. K. Kraft, and J. L. Farber. The nephrotoxicity of vancomycin. *Clinical pharmacology and therapeutics*, 102(3):459–469, Sep 2017. 28474732[pmid]. 31

[80] C. Lei, L. Berra, E. Rezoagli, et al. Nitric oxide decreases acute kidney injury and stage 3 chronic kidney disease after cardiac surgery. *American Journal of Respiratory and Critical Care Medicine*, 198(10):1279–1287, 2018. 31

[81] S. Farhan, B. Vogel, U. Baber, et al. Calculated serum osmolality, acute kidney injury, and relationship to mortality after percutaneous coronary intervention. *Cardiorenal Med*, 9(3):160–167, 2019. 31

[82] K. Hahn, M. Kanbay, M. A. Lanaspa, R. J. Johnson, and Ejaz AA. Serum uric acid and acute kidney injury: A mini review. *Journal of Advanced Research*, 8(5):529–536, 2016. 31

[83] MIT Laboratory for Computational Physiology. Mimic code repository. https://github.com/MIT-LCP/mimic-code. 32

[84] Jason Brownlee. How to calculate feature importance with python. https://machinelearningmastery.com/calculate-feature-importance-with-python/, Aug 2020. Machine Learning Mastery. 39

[85] Piotr Płoński. Random forest feature importance computed in 3 ways with python. https://mljar.com/blog/feature-importance-in-random-forest/, Jun 2020. mljar. 41

[86] Joos Korstanje. Polynomial interpolation. Retrieved from https://towardsdatascience.com/polynomial-interpolation-3463ea4b63dd, Jun 2021. Medium. 67

Appendix A

List of the Variables Extracted

* a) MIMIC-III Community; b) Manually selected c) Calculated

Variable	Label	Origin*	Units
Time Series			
1	Alanine aminotransferase (ALT)	a)	[IU/L]
2	Alkaline phosphatase (ALP)	a)	[IU/L]
3	Anion Gap (AG)	a)	[mmol/L]
4	Arterial Base Excess (aBE)	b)	[mmol/L]
5	Arterial CO_2	b)	[mmol/L]
6	Arterial Partial Pressure CO_2 (PaCO2)	a)	[mmHg]
7	Arterial Partial Pressure O_2 (PaO2)	a)	[mmHg]
8	Arterial pH	a)	No units
9	Asparate Aminotransferase (AST)	a)	[IU/L]
10	Basophils	b)	[x10^9 cells/L]
11	Bicarbonate	a)	[mmol/L]
12	Bilirubin (BR)	a)	[mg/dL]
13	Blood urea nitrogen (BUN)	a)	[mg/dL]

14	Calcium	b)	[mg/dL]
15	Central Venous Pressure (CVP)	b)	[mmHg]
16	Chloride	a)	[mmol/L]
17	Creatine Phosphokinase (CPK)	b)	[U/L]
18	Creatinine - Baseline value (Creat)	c)	[mg/dL]
19	Creatinine - Baseline value: Lowest 48hr (Creat48h)	c)	[mg/dL]
20	Creatinine - Baseline value: Lowest 7 days (Creat7d)	c)	[mg/dL]
21	Diastolic Blood Pressure (D-BP)	a)	[mmHg]
22	Glasgow Coma Scale Total (GCS)	b)	[3-15]
23	Glucose	a)	[mg/dL]
24	Heart Rate (HR)	a)	[beats pm]
25	Hematocrit	a)	[%]
26	Hemoglobin	a)	[g/dL]
27	Lactate	a)	[mg/dL]
28	Lactic Acid	b)	[mmol/L]
29	Magnesium	a)	[mg/dL]
30	Mean Blood Pressure (M-BP)	a)	[mmHg]
31	Mean Corpuscular Hemoglobin (MCH)	b)	[pg/cell]
32	Mean Corpuscular Hemoglobin Concentration (MCHC)	b)	[g/dL]
33	Mean Corpuscular Volume (MCV)	b)	[fL]

34	Partial Thromboplastin Time (PTT)	a)	[s]
35	Peak Inspiratory Pressure (PIP)	a)	[cmH2O]
36	Phosphate	a)	[mmol/L]
37	Phosphorous	b)	[mmol/L]
38	Platelets	a)	[$x10^9$/L]
39	Positive end-expiratory pressure (PEEP)	a)	[cmH2O]
40	Potassium	a)	[mmol/L]
41	Prothrombin Time (PT)	a)	[s]
42	Red Blood Cell Count (RBC)	b)	[$x10^{12}$ cells/L]
43	Red Blood Cell Distribution Width (RDW)	b)	[%]
44	Respiratory Rate (RR)	a)	[breaths pm]
45	Saturation of O_2 (SpO2)	a)	[%]
46	Serum Albumin (ALB)	a)	[g/dL]
47	Serum Creatinine (sCr)	a)	[mg/dL]
48	Sodium	a)	[mmol/L]
49	Systolic Blood Pressure (S-BP)	a)	[mmHg]
50	Temperature	a)	[ºC]
51	Urine Output Rate: 6hr (UO6hr)	c)	[mL]
52	Urine Output Rate: 12hr (UO12hr)	c)	[mL]
53	Urine Output Rate: 24hr (UO24hr)	c)	[mL]
54	Venous Base Excess (vBE)	b)	[mmol/L]

55	Venous Partial Pressure O_2 (PvO2)	b)	[mmHg]
56	Weight	a)	[kg]
57	White Blood Cells Count (WBC)	a)	[x10^9 cells/L]
58	AKI Stage	c)	[0 to 3]
Categorical Variables			
59	Ectopy Frequency	b)	None = 0 / Rare = 1 / Occasional = 2 / Frequent = 3 / Has Ventricular Tachycardia(V-Tach) = 4]
60	GCS Eye Opening	b)	[No Response = 1 / To Pain = 2 / To Speech = 3 / Spontaneously = 4]
61	GCS Motor Response	b)	[No Response = 1 / Abnormal Extension = 2 / Abnormal Flexion = 3/ Flex to withdraw from pain = 4 / Moves to localize pain = 5 / Obeys commands = 6]
62	GCS Verbal Response	b)	[No Response = 1 / Incomprehensible sounds = 2 / Inappropriate words = 3 / Confused = 4 / Oriented to time, person and place = 5]
63	Urine Appearance	b)	[Clear = 1 / Cloudy = 2 / Sediment = 3 / Sludge = 4 / Clots = 5]
64	Urine Color	b)	[Light Yellow = 1 / Yellow= 2 / Icteric = 3 / Amber = 4 /Orange = 5 / Pink = 5 / Red = 6 / Brown = 7]

Appendix B

Feature Importance Scores for all patients

Table B.1: Feature Importance Scores using all features with 2B

Feature	Score
AKI Stage	0.285009
Urine Output Rate: 12hr	0.184183
Urine Output Rate: 6hr	0.173600
Urine Output Rate: 24hr	0.114086
Systolic Blood Pressure	0.010027
Mean Blood Pressure	0.009850
Heart Rate	0.009611
Diastolic Blood Pressure	0.009301
Respiratory Rate	0.008150
Temperature	0.007122
Weight	0.006702
Glucose	0.006430
Saturation of O2	0.006082
Urine Color	0.005734
Central Venous Pressure	0.005217
Arterial Partial Pressure O2	0.004377
Prothrombin Time	0.004299
Hematocrit	0.004236
Platelets	0.004192
Partial Thromboplastin Time	0.004140
Creatinine - Baseline value	0.004078
Red Blood Cell Count (RBC)	0.004018
Serum Creatinine	0.003989
Peak Inspiratory Pressure	0.003982
White Blood Cells Count (WBC)	0.003961
Arterial pH	0.003932
Asparate Aminotransferase	0.003911
Red Blood Cell Distribution Width (RDW)	0.003893
Venous Partial Pressure O2	0.003787

Table B.1 continued from previous page

Blood urea nitrogen	0.003745
Bicarbonate	0.00372
Mean Corpuscular Hemoglobin	0.00367
Mean Corpuscular Hemoglobin Concentration	0.00367
Hemoglobin	0.00363
Arterial Partial Pressure CO2	0.003601
Alkaline phosphatase	0.003597
Arterial CO2	0.003526
Potassium	0.003472
Venous Base Excess	0.003454
Arterial Base Excess	0.003445
Phosphorous	0.003397
Calcium	0.003332
Phosphate	0.003329
Sodium	0.003256
Chloride	0.003216
Alanine aminotransferase	0.003198
Mean Corpuscular Volume	0.003168
Creatine Phosphokinase (CPK)	0.003160
Creatinine - Baseline value: Lowest 7 days	0.003158
Lactic Acid	0.003023
Lactate	0.002995
Serum Albumin	0.002984
Magnesium	0.002938
Bilirubin	0.002923
Creatinine - Baseline value: Lowest 48hr	0.002813
Glasgow Coma Scale Total	0.002711
Anion Gap	0.002630
Urine Appearance	0.001984
Positive end-expiratory pressure (PEEP)	0.001892
Ectopy Frequency	0.001582
Basophils	0.001438
Glasgow Coma Scale Eye Opening	0.001430
Glasgow Coma Scale Motor Response	0.001292
Glasgow Coma Scale Verbal Response	0.000719

Table B.2: Feature Importance Scores using no calculated features with 2B

Feature	Score
Weight	0.036468
Heart Rate	0.035039
Temperature	0.032072
Systolic Blood Pressure	0.030230
Mean Blood Pressure	0.029822

Table B.2 continued from previous page

Glucose	0.029752
Diastolic Blood Pressure	0.028170
Respiratory Rate	0.023741
Urine Color	0.022156
Platelets	0.021321
Central Venous Pressure	0.020664
Peak Inspiratory Pressure	0.020631
Prothrombin Time	0.020490
Mean Corpuscular Hemoglobin	0.019693
Saturation of O2	0.019533
Asparate Aminotransferase	0.019525
Serum Creatinine	0.019330
Blood urea nitrogen	0.019044
White Blood Cells Count (WBC)	0.018950
Partial Thromboplastin Time	0.018813
Red Blood Cell Distribution Width (RDW)	0.018712
Hematocrit	0.018694
Red Blood Cell Count (RBC)	0.018552
Alkaline phosphatase	0.018162
Mean Corpuscular Hemoglobin Concentration	0.017444
Arterial Partial Pressure O2	0.017339
Arterial pH	0.016800
Alanine aminotransferase	0.016478
Bicarbonate	0.015995
Mean Corpuscular Volume	0.015954
Hemoglobin	0.015944
Creatine Phosphokinase (CPK)	0.015893
Venous Partial Pressure O2	0.015797
Arterial Partial Pressure CO2	0.015781
Potassium	0.015719
Phosphorous	0.015407
Arterial CO2	0.015314
Calcium	0.014912
Phosphate	0.014852
Chloride	0.014711
Serum Albumin	0.014453
Sodium	0.014298
Bilirubin	0.014281
Venous Base Excess	0.014057
Lactate	0.013963
Lactic Acid	0.013531
Glasgow Coma Scale Total	0.013307
Arterial Base Excess	0.013294
Anion Gap	0.013241

Table B.2 continued from previous page

Magnesium	0.013149
Positive end-expiratory pressure (PEEP)	0.010307
Urine Appearance	0.008413
Glasgow Coma Scale Eye Opening	0.007872
Basophils	0.006697
Glasgow Coma Scale Motor Response	0.006395
Ectopy Frequency	0.005554
Glasgow Coma Scale Verbal Response	0.003288

Table B.3: Feature Importance Scores using all features with sCr

Feature	Score
AKI Stage	0.509471
Creatinine - Baseline value	0.063913
Creatinine - Baseline value: Lowest 7 days	0.057625
Creatinine - Baseline value: Lowest 48hr	0.038382
Serum Creatinine	0.023816
Creatine Phosphokinase (CPK)	0.022658
Weight	0.020595
Bilirubin	0.020299
Partial Thromboplastin Time	0.016286
Alkaline phosphatase	0.011447
Asparate Aminotransferase	0.011236
Alanine aminotransferase	0.010826
Blood urea nitrogen	0.010374
Serum Albumin	0.010319
Prothrombin Time	0.010145
Mean Corpuscular Hemoglobin	0.008142
Mean Corpuscular Volume	0.007800
Red Blood Cell Distribution Width (RDW)	0.007788
Platelets	0.006803
Lactate	0.006603
Lactic Acid	0.006301
Peak Inspiratory Pressure	0.006122
Basophils	0.005681
Red Blood Cell Count (RBC)	0.005178
Urine Output Rate: 24hr	0.005169
White Blood Cells Count (WBC)	0.005112
Phosphorous	0.004776
Phosphate	0.004551
Arterial Partial Pressure CO2	0.004392
Hemoglobin	0.004377
Central Venous Pressure	0.004100
Positive end-expiratory pressure(PEEP)	0.004030

Table B.3 continued from previous page

Arterial CO2	0.003922
Hematocrit	0.003844
Mean Corpuscular Hemoglobin Concentration	0.003742
Bicarbonate	0.003629
Chloride	0.003495
Sodium	0.003392
Calcium	0.003143
Urine Output Rate: 12hr	0.002953
Anion Gap	0.002757
Arterial pH	0.002756
Arterial Base Excess	0.002606
Venous Partial Pressure O2	0.002549
Magnesium	0.002497
Urine Output Rate: 6hr	0.002299
Arterial Partial Pressure O2	0.002223
Heart Rate	0.002089
Venous Base Excess	0.002000
Glasgow Coma Scale Total	0.001831
Potassium	0.001663
Temperature	0.001493
Diastolic Blood Pressure	0.001365
Mean Blood Pressure	0.001275
Systolic Blood Pressure	0.001263
Glucose	0.001204
Glasgow Coma Scale Motor Response	0.001042
Respiratory Rate	0.001021
Saturation of O2	0.000738
Glasgow Coma Scale Eye Opening	0.000723
Ectopy Frequency	0.000644
Urine Color	0.000504
Urine Appearance	0.000306
Glasgow Coma Scale Verbal Response	0.000279

Table B.4: Feature Importance Scores using no calculated features with sCr

Feature	Score
Serum Creatinine	0.057755
Weight	0.057478
Bilirubin	0.053069
Creatine Phosphokinase (CPK)	0.049827
Partial Thromboplastin Time	0.041221
Alkaline phosphatase	0.033444
Asparate Aminotransferase	0.033382
Alanine aminotransferase	0.032885

Table B.4 continued from previous page

Prothrombin Time	0.032151
Serum Albumin	0.031165
Blood urea nitrogen	0.030761
Red Blood Cell Distribution Width (RDW)	0.026779
Platelets	0.024591
Mean Corpuscular Volume	0.024179
Mean Corpuscular Hemoglobin	0.022583
Phosphorous	0.020293
Lactic Acid	0.020137
Lactate	0.019367
Peak Inspiratory Pressure	0.018737
Red Blood Cell Count (RBC)	0.018417
White Blood Cells Count (WBC)	0.017879
Arterial Partial Pressure CO2	0.016860
Chloride	0.016858
Basophils	0.016698
Phosphate	0.016553
Sodium	0.016069
Hematocrit	0.015987
Hemoglobin	0.015889
Mean Corpuscular Hemoglobin Concentration	0.015295
Central Venous Pressure	0.015123
Bicarbonate	0.015027
Arterial pH	0.013720
Calcium	0.013366
Positive end-expiratory pressure (PEEP)	0.013206
Arterial CO2	0.012044
Magnesium	0.011796
Anion Gap	0.011688
Venous Partial Pressure O2	0.010205
Arterial Base Excess	0.009633
Arterial Partial Pressure O2	0.009102
Venous Base Excess	0.009097
Potassium	0.008179
Heart Rate	0.007245
Glasgow Coma Scale Total	0.005844
Temperature	0.005213
Glucose	0.004956
Glasgow Coma Scale Motor Response	0.004314
Diastolic Blood Pressure	0.003189
Systolic Blood Pressure	0.003035
Respiratory Rate	0.002946
Mean Blood Pressure	0.002922
Glasgow Coma Scale Eye Opening	0.002644

Table B.4 continued from previous page

Urine Color	0.002378
Glasgow Coma Scale Verbal Response	0.001812
Saturation of O2	0.001787
Ectopy Frequency	0.001765
Urine Appearance	0.001453

Table B.5: Feature Importance Scores using all features with 2B Raw

Feature	Score
AKI Stage	0.227644
Urine Output Rate: 12hr	0.197727
Urine Output Rate: 6hr	0.185722
Urine Output Rate: 24hr	0.109157
Systolic Blood Pressure	0.012061
Mean Blood Pressure	0.011935
Heart Rate	0.011678
Diastolic Blood Pressure	0.011305
Respiratory Rate	0.009717
Temperature	0.008277
Glucose	0.007528
Saturation of O2	0.007377
Weight	0.007361
Central Venous Pressure	0.006238
Urine Color	0.005422
Arterial Partial Pressure O2	0.005138
Platelets	0.004842
Partial Thromboplastin Time	0.004842
Peak Inspiratory Pressure	0.004718
Hematocrit	0.004666
White Blood Cells Count (WBC)	0.004601
Serum Creatinine	0.004522
Venous Partial Pressure O2	0.004514
Arterial pH	0.004458
Prothrombin Time	0.004416
Mean Corpuscular Hemoglobin	0.004384
Mean Corpuscular Hemoglobin Concentration	0.004349
Arterial Partial Pressure CO2	0.004341
Red Blood Cell Count (RBC)	0.004329
Asparate Aminotransferase	0.004316
Creatinine - Baseline value	0.004304
Red Blood Cell DistributionWidth (RDW)	0.004283
Blood urea nitrogen	0.004191
Alkaline phosphatase	0.004182
Arterial CO2	0.004148

Table B.5 continued from previous page

Hemoglobin	0.004080
Phosphorous	0.004066
Phosphate	0.004065
Potassium	0.004058
Calcium	0.003986
Creatinine - Baseline value: Lowest 7 days	0.003854
Bicarbonate	0.003791
Chloride	0.003700
Creatine Phosphokinase (CPK)	0.003698
Sodium	0.003689
Alanine aminotransferase	0.003665
Mean Corpuscular Volume	0.003591
Venous Base Excess	0.003561
Anion Gap	0.003551
Magnesium	0.003518
Lactate	0.003498
Arterial Base Excess	0.003487
Serum Albumin	0.003412
Lactic Acid	0.003333
Glasgow Coma Scale Total	0.003190
Bilirubin	0.003149
Creatinine - Baseline value: Lowest 48hr	0.003094
Positive end-expiratory pressure (PEEP)	0.001983
Glasgow Coma Scale Eye Opening	0.001825
Ectopy Frequency	0.001748
Urine Appearance	0.001748
Basophils	0.001716
Glasgow Coma Scale Motor Response	0.001416
Glasgow Coma Scale Verbal Response	0.000855

Table B.6: Feature Importance Scores using no calculated features with 2B Raw

Feature	Score
Heart Rate	0.037282
Weight	0.036084
Systolic Blood Pressure	0.033101
Temperature	0.033073
Mean Blood Pressure	0.032924
Diastolic Blood Pressure	0.031078
Glucose	0.030213
Respiratory Rate	0.025866
Saturation of O2	0.021169
Platelets	0.020940
Central Venous Pressure	0.020927

Table B.6 continued from previous page

Urine Color	0.020663
Peak Inspiratory Pressure	0.020486
Prothrombin Time	0.020431
Mean Corpuscular Hemoglobin	0.019391
White Blood Cells Count (WBC)	0.018837
Partial Thromboplastin Time	0.018723
Asparate Aminotransferase	0.018590
Blood urea nitrogen	0.018461
Serum Creatinine	0.018349
Hematocrit	0.018268
Red Blood Cell Distribution Width (RDW)	0.018169
Red Blood Cell Count (RBC)	0.017755
Alkaline phosphatase	0.017416
Mean Corpuscular Hemoglobin Concentration	0.017411
Arterial Partial Pressure O2	0.017098
Arterial pH	0.016521
Alanine aminotransferase	0.015960
Hemoglobin	0.015871
Venous Partial Pressure O2	0.015811
Potassium	0.015741
Mean Corpuscular Volume	0.015435
Arterial Partial Pressure CO2	0.015432
Bicarbonate	0.015371
Arterial CO2	0.015330
Phosphate	0.014812
Phosphorous	0.014812
Calcium	0.014693
Chloride	0.014471
Creatine Phosphokinase (CPK)	0.014470
Serum Albumin	0.014388
Bilirubin	0.014080
Sodium	0.013946
Lactate	0.013655
Lactic Acid	0.013537
Glasgow Coma Scale Total	0.013449
Anion Gap	0.013208
Magnesium	0.012980
Arterial Base Excess	0.012905
Venous Base Excess	0.012823
Positive end-expiratory pressure (PEEP)	0.009889
Urine Appearance	0.007954
Glasgow Coma Scale Eye Opening	0.007868
Basophils	0.006655
Glasgow Coma Scale Motor Response	0.006157

Table B.6 continued from previous page

Ectopy Frequency	0.005646
Glasgow Coma Scale Verbal Response	0.003428

Appendix C

Feature Importance Scores for the reduced patient cohort

Table C.1: Feature Importance Scores using all features with 2B

Feature	Score
AKI Stage	0.264740
Urine Output Rate: 12hr	0.166912
Urine Output Rate: 6hr	0.157392
Urine Output Rate: 24hr	0.134967
Heart Rate	0.010969
Systolic Blood Pressure	0.010804
Mean Blood Pressure	0.010038
Diastolic Blood Pressure	0.009764
Respiratory Rate	0.008857
Temperature	0.008564
Glucose	0.007815
Urine Color	0.007566
Arterial pH	0.007376
Saturation of O2	0.006699
Blood urea nitrogen	0.006169
Central Venous Pressure	0.005662
Bicarbonate	0.005526
Hematocrit	0.005277
Platelets	0.005248
Peak Inspiratory Pressure	0.005140
Asparate Aminotransferase	0.005080
Arterial CO2	0.005070
Arterial Partial Pressure O2	0.004906
Red Blood Cell Count (RBC)	0.004742
Partial Thromboplastin Time	0.004686
White Blood Cells Count (WBC)	0.004602
Serum Creatinine	0.004589

Table C.1 continued from previous page

Venous Partial Pressure O2	0.004389
Creatinine - Baseline value	0.004377
Mean Corpuscular Hemoglobin	0.004326
Mean Corpuscular Hemoglobin Concentration	0.004250
Potassium	0.004220
Hemoglobin	0.004120
Phosphorous	0.004010
Phosphate	0.003977
Red Blood Cell Distribution Width (RDW)	0.003951
Prothrombin Time	0.003950
Lactate	0.003949
Arterial Base Excess	0.003931
Chloride	0.003773
Arterial Partial Pressure CO2	0.003769
Calcium	0.003762
Venous Base Excess	0.003658
Glasgow Coma Scale Total	0.003609
Alanine aminotransferase	0.003601
Bilirubin	0.003586
Anion Gap	0.003563
Mean Corpuscular Volume	0.003533
Weight	0.003502
Sodium	0.003482
Creatinine - Baseline value: Lowest 48hr	0.003228
Alkaline phosphatase	0.003195
Creatine Phosphokinase (CPK)	0.003116
Magnesium	0.003058
Lactic Acid	0.002971
Serum Albumin	0.002772
Positive end-expiratory pressure (PEEP)	0.002052
Urine Appearance	0.002014
Creatinine - Baseline value: Lowest 7 days	0.001964
Glasgow Coma Scale Eye Opening	0.001913
Ectopy Frequency	0.001665
Basophils	0.001640
Glasgow Coma Scale Motor Response	0.001120
Glasgow Coma Scale Verbal Response	0.000854

Table C.2: Feature Importance Scores using no calculated features with 2B

Feature	Score
Temperature	0.036204
Heart Rate	0.035290
Glucose	0.033326

Table C.2 continued from previous page

Systolic Blood Pressure	0.031361
Mean Blood Pressure	0.029542
Diastolic Blood Pressure	0.028462
Blood urea nitrogen	0.027602
Urine Color	0.025182
Respiratory Rate	0.024391
Platelets	0.023244
Peak Inspiratory Pressure	0.023199
Saturation of O2	0.021125
Central Venous Pressure	0.020874
Arterial pH	0.019868
White Blood Cells Count (WBC)	0.019138
Hematocrit	0.018958
Arterial Partial Pressure O2	0.018772
Red Blood Cell Distribution Width (RDW)	0.018659
Venous Partial Pressure O2	0.018150
Bicarbonate	0.017927
Partial Thromboplastin Time	0.017686
Arterial CO2	0.017393
Phosphorous	0.017287
Potassium	0.017275
Mean Corpuscular Hemoglobin Concentration	0.017231
Serum Creatinine	0.017169
Arterial Partial Pressure CO2	0.017095
Red Blood Cell Count (RBC)	0.016889
Asparate Aminotransferase	0.016886
Sodium	0.016690
Mean Corpuscular Hemoglobin	0.016468
Prothrombin Time	0.016375
Glasgow Coma Scale Total	0.016290
Hemoglobin	0.015968
Calcium	0.015961
Weight	0.015610
Phosphate	0.015581
Chloride	0.015317
Alanine aminotransferase	0.014652
Creatine Phosphokinase (CPK)	0.014222
Bilirubin	0.014217
Lactate	0.014153
Mean Corpuscular Volume	0.013933
Magnesium	0.013121
Arterial Base Excess	0.012962
Anion Gap	0.012930
Venous Base Excess	0.012817

Table C.2 continued from previous page

Lactic Acid	0.012155
Alkaline phosphatase	0.011384
Serum Albumin	0.010779
Urine Appearance	0.010024
Positive end-expiratory pressure (PEEP)	0.009550
Glasgow Coma Scale Eye Opening	0.008938
Basophils	0.007606
Glasgow Coma Scale Motor Response	0.006256
Ectopy Frequency	0.005593
Glasgow Coma Scale Verbal Response	0.004259

Table C.3: Feature Importance Scores using all features with sCr

Feature	Score
AKI Stage	0.334449
Creatinine - Baseline value	0.102637
Creatinine - Baseline value: Lowest 7 days	0.061051
Creatinine - Baseline value: Lowest 48hr	0.050543
Weight	0.035537
Serum Creatinine	0.034810
Creatine Phosphokinase (CPK)	0.024888
Red Blood Cell Distribution Width (RDW)	0.019818
Blood urea nitrogen	0.018374
Alkaline phosphatase	0.017137
Prothrombin Time	0.015007
Partial Thromboplastin Time	0.014949
Asparate Aminotransferase	0.014335
Bilirubin	0.013154
Mean Corpuscular Volume	0.012559
Alanine aminotransferase	0.012298
Peak Inspiratory Pressure	0.011178
Serum Albumin	0.011148
Hematocrit	0.010028
Lactic Acid	0.010028
Lactate	0.009813
Red Blood Cell Count (RBC)	0.009645
Platelets	0.009371
Mean Corpuscular Hemoglobin	0.009158
White Blood Cells Count (WBC)	0.008658
Hemoglobin	0.007803
Phosphorous	0.007128
Central Venous Pressure	0.007084
Phosphate	0.006823
Bicarbonate	0.005894

Table C.3 continued from previous page

Positive end-expiratory pressure (PEEP)	0.005890
Anion Gap	0.005290
Urine Output Rate: 24hr	0.005250
Basophils	0.004930
Mean Corpuscular Hemoglobin Concentration	0.004756
Chloride	0.004663
Arterial Partial Pressure CO2	0.004642
Arterial CO2	0.004397
Calcium	0.004349
Sodium	0.004310
Urine Output Rate: 12hr	0.003622
Venous Base Excess	0.003080
Magnesium	0.003014
Arterial Base Excess	0.002947
Heart Rate	0.002708
Potassium	0.002609
Arterial Partial Pressure O2	0.002601
Glasgow Coma Scale Total	0.002453
Urine Output Rate: 6hr	0.002408
Temperature	0.002241
Venous Partial Pressure O2	0.002196
Arterial pH	0.002156
Glasgow Coma Scale Motor Response	0.001724
Diastolic Blood Pressure	0.001719
Glucose	0.001658
Systolic Blood Pressure	0.001641
Mean Blood Pressure	0.001472
Glasgow Coma Scale Eye Opening	0.001277
Respiratory Rate	0.001124
Urine Color	0.000862
Saturation of O2	0.000805
Glasgow Coma Scale Verbal Response	0.000756
Urine Appearance	0.000635
Ectopy Frequency	0.000524

Table C.4: Feature Importance Scores using no calculated features with sCr

Feature	Score
Serum Creatinine	0.076758
Weight	0.052745
Creatine Phosphokinase (CPK)	0.051706
Blood urea nitrogen	0.046112
Alkaline phosphatase	0.040331
Partial Thromboplastin Time	0.034188

Table C.4 continued from previous page

Red Blood Cell Distribution Width (RDW)	0.033814
Bilirubin	0.031066
Asparate Aminotransferase	0.028934
Alanine aminotransferase	0.028625
Platelets	0.028054
Prothrombin Time	0.027635
Serum Albumin	0.027199
Mean Corpuscular Volume	0.026977
Red Blood Cell Count (RBC)	0.023519
Lactic Acid	0.023180
Lactate	0.022922
Peak Inspiratory Pressure	0.022485
Mean Corpuscular Hemoglobin	0.022444
White Blood Cells Count (WBC)	0.021264
Hematocrit	0.020658
Phosphate	0.019123
Hemoglobin	0.018775
Phosphorous	0.018005
Basophils	0.016668
Arterial Partial Pressure CO2	0.014956
Chloride	0.013670
Positive end-expiratory pressure (PEEP)	0.013588
Central Venous Pressure	0.013316
Bicarbonate	0.013050
Sodium	0.012976
Mean Corpuscular Hemoglobin Concentration	0.012720
Arterial CO2	0.012454
Calcium	0.011236
Anion Gap	0.010744
Arterial Base Excess	0.009594
Magnesium	0.008710
Venous Base Excess	0.007829
Arterial pH	0.007688
Glasgow Coma Scale Total	0.007322
Arterial Partial Pressure O2	0.007069
Potassium	0.006981
Venous Partial Pressure O2	0.006937
Heart Rate	0.006468
Temperature	0.004612
Glasgow Coma Scale Motor Response	0.004604
Glucose	0.004155
Glasgow Coma Scale Eye Opening	0.003798
Mean Blood Pressure	0.003501
Diastolic Blood Pressure	0.003462

Table C.4 continued from previous page

Systolic Blood Pressure	0.003112
Respiratory Rate	0.002777
Urine Color	0.002401
Glasgow Coma Scale Verbal Response	0.002349
Urine Appearance	0.001791
Saturation of O2	0.001543
Ectopy Frequency	0.001396

Table C.5: Feature Importance Scores using all features with 2B Raw

Feature	Score
AKI Stage	0.251922
Urine Output Rate: 12hr	0.190764
Urine Output Rate: 24hr	0.129226
Urine Output Rate: 6hr	0.124667
Heart Rate	0.013305
Systolic Blood Pressure	0.012967
Mean Blood Pressure	0.012088
Diastolic Blood Pressure	0.011896
Respiratory Rate	0.010440
Temperature	0.009451
Glucose	0.008919
Saturation of O2	0.008252
Urine Color	0.007649
Central Venous Pressure	0.006685
Arterial pH	0.006645
Peak Inspiratory Pressure	0.005675
Arterial Partial Pressure O2	0.005471
Platelets	0.005375
Blood urea nitrogen	0.005355
Bicarbonate	0.005288
White Blood Cells Count (WBC)	0.005234
Hematocrit	0.005107
Partial Thromboplastin Time	0.005072
Arterial CO2	0.005039
Potassium	0.005036
Red Blood Cell Count (RBC)	0.005006
Serum Creatinine	0.004893
Venous Partial Pressure O2	0.004821
Hemoglobin	0.004729
Red Blood Cell Distribution Width (RDW)	0.004634
Phosphorous	0.00455
Asparate Aminotransferase	0.00452
Phosphate	0.00448

Table C.5 continued from previous page

Sodium	0.00446
Chloride	0.004344
Calcium	0.004344
Arterial Partial Pressure CO2	0.004303
Bilirubin	0.004293
Mean Corpuscular Hemoglobin Concentration	0.004263
Prothrombin Time	0.004216
Mean Corpuscular Hemoglobin	0.004211
Weight	0.004068
Creatinine - Baseline value	0.004059
Venous Base Excess	0.004025
Mean Corpuscular Volume	0.004008
Glasgow Coma Scale Total	0.003986
Alanine aminotransferase	0.003862
Arterial Base Excess	0.003728
Creatinine - Baseline value: Lowest 48hr	0.003701
Alkaline phosphatase	0.003648
Lactate	0.003600
Anion Gap	0.003553
Lactic Acid	0.003455
Magnesium	0.003348
Serum Albumin	0.003339
Creatine Phosphokinase (CPK)	0.003240
Positive end-expiratory pressure (PEEP)	0.002448
Creatinine - Baseline value: Lowest 7 days	0.002324
Urine Appearance	0.002051
Glasgow Coma Scale Eye Opening	0.002011
Ectopy Frequency	0.001895
Basophils	0.001658
Glasgow Coma Scale Motor Response	0.001433
Glasgow Coma Scale Verbal Response	0.000964

Table C.6: Feature Importance Scores using no calculated features with 2B Raw

Feature	Score
Heart Rate	0.037868
Temperature	0.036524
Glucose	0.034735
Systolic Blood Pressure	0.034343
Mean Blood Pressure	0.032548
Diastolic Blood Pressure	0.030956
Respiratory Rate	0.026172
Blood urea nitrogen	0.023179
Peak Inspiratory Pressure	0.022729

Table C.6 continued from previous page

Saturation of O2	0.022611
Platelets	0.022517
Urine Color	0.021889
Central Venous Pressure	0.021441
Arterial pH	0.019879
Hematocrit	0.019134
Red Blood Cell Distribution Width (RDW)	0.018949
White Blood Cells Count (WBC)	0.018867
Arterial Partial Pressure O2	0.018542
Prothrombin Time	0.018244
Venous Partial Pressure O2	0.018128
Arterial Partial Pressure CO2	0.017821
Partial Thromboplastin Time	0.017682
Mean Corpuscular Hemoglobin Concentration	0.017638
Potassium	0.017577
Arterial CO2	0.017524
Red Blood Cell Count (RBC)	0.017076
Bicarbonate	0.017037
Serum Creatinine	0.016908
Mean Corpuscular Hemoglobin	0.016883
Calcium	0.016830
Sodium	0.016662
Hemoglobin	0.016651
Weight	0.015982
Chloride	0.015707
Phosphorous	0.015644
Glasgow Coma Scale Total	0.015634
Phosphate	0.015092
Bilirubin	0.013995
Asparate Aminotransferase	0.013698
Mean Corpuscular Volume	0.013624
Creatine Phosphokinase (CPK)	0.013568
Lactate	0.012952
Alanine aminotransferase	0.012763
Magnesium	0.012393
Arterial Base Excess	0.012139
Anion Gap	0.012122
Lactic Acid	0.012080
Alkaline phosphatase	0.011748
Venous Base Excess	0.011706
Serum Albumin	0.011464
Positive end-expiratory pressure (PEEP)	0.009878
Urine Appearance	0.009735
Glasgow Coma Scale Eye Opening	0.009390

Table C.6 continued from previous page

Basophils	0.006552
Ectopy Frequency	0.006218
Glasgow Coma Scale Motor Response	0.006004
Glasgow Coma Scale Verbal Response	0.004368

Appendix D

Results for All Patients

Table D.1: 2B: Predicting the current hour (mean ± standard deviation)

		Number of features								
		5	10	20	30	40	50	57 (All)	60	63 (All)
Full Dataset (No Stage)	RF	0.9914 ± 0.0003	0.9909 ± 0.0005	0.9919 ± 0.0008	0.9924 ± 0.0006	0.9925 ± 0.0004	0.9924 ± 0.0006	-	0.9925 ± 0.0007	0.9926 ± 0.0006
	NB	0.9039 ± 0.0016	0.9013 ± 0.0018	0.8996 ± 0.0021	0.8944 ± 0.0022	0.8915 ± 0.0025	0.8865 ± 0.0028	-	0.8824 ± 0.0028	0.8816 ± 0.0030
No Stage & Baselines	RF	0.8852 ± 0.0018	0.9119 ± 0.0013	0.9378 ± 0.0013	0.9439 ± 0.0016	0.9452 ± 0.0017	0.9467 ± 0.0014	0.9476 ± 0.0012	-	-
	NB	0.8191 ± 0.0015	0.8057 ± 0.0014	0.7842 ± 0.0021	0.7764 ± 0.0031	0.7674 ± 0.0030	0.7471 ± 0.0025	0.7395 ± 0.0022	-	-

Table D.2: 2B Raw: Predicting the current hour (mean ± standard deviation)

		Number of features								
		5	10	20	30	40	50	57 (All)	60	63 (All)
Full Dataset (No Stage)	RF	0.9987 ± 0.0011	0.9881 ± 0.0012	0.9900 ± 0.0012	0.9904 ± 0.0011	0.9904 ± 0.0010	0.9903 ± 0.0010	-	0.9902 ± 0.0011	0.9902 ± 0.0011
	NB	0.7938 ± 0.0045	0.7958 ± 0.0041	0.7964 ± 0.0029	0.7952 ± 0.0025	0.7937 ± 0.0031	0.7984 ± 0.0024	-	0.7950 ± 0.0027	0.7948 ± 0.0032
No Stage & Baselines	RF	0.8590 ± 0.0011	0.8693 ± 0.0011	0.9286 ± 0.0015	0.9349 ± 0.0014	0.9361 ± 0.0016	0.9371 ± 0.0022	0.9390 ± 0.0016	-	-
	NB	0.8156 ± 0.0017	0.8101 ± 0.0018	0.7809 ± 0.0023	0.7733 ± 0.0036	0.7657 ± 0.0037	0.7447 ± 0.0038	0.7373 ± 0.0035	-	-

Table D.3: Creat: Predicting the current hour (mean ± standard deviation)

		Number of features								
		5	10	20	30	40	50	57 (All)	60	63 (All)
Full Dataset (No Stage)	RF	0.9999 ± 0.0000	1.0 ± 0.0000	0.9999 ± 0.0000	0.9999 ± 0.0000	0.9999 ± 0.0000	0.9999 ± 0.0000	-	0.9999 ± 0.0000	0.9999 ± 0.0000
	NB	0.7532 ± 0.0024	0.7267 ± 0.0031	0.7366 ± 0.0036	0.7258 ± 0.0041	0.7148 ± 0.0033	0.7009 ± 0.0041	-	0.6996 ± 0.0037	0.6983 ± 0.0039
No Stage & Baselines	RF	0.9992 ± 0.0003	0.9999 ± 0.0000	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0001	-	-
	NB	0.7668 ± 0.0024	0.7445 ± 0.0022	0.7342 ± 0.0028	0.7258 ± 0.0035	0.7060 ± 0.0035	0.7044 ± 0.0035	0.6951 ± 0.0034	-	-

Table D.4: 2B: Predicting the next hour (mean ± standard deviation)

		Number of features									
		5	10	20	30	40	50	59 (All)	60	65 (All)	66 (All)
Full Dataset	RF	0.9573 ± 0.0013	0.9610 ± 0.0016	0.9595 ± 0.0015	0.9579 ± 0.0015	0.9570 ± 0.0015	0.9563 ± 0.0014	-	0.9558 ± 0.0017	-	0.9561 ± 0.0017
	NB	0.9475 ± 0.0024	0.9268 ± 0.0028	0.9239 ± 0.0024	0.9173 ± 0.0028	0.9126 ± 0.0029	0.9054 ± 0.0031	-	0.8953 ± 0.0034	-	0.8942 ± 0.0032
Full Dataset (No Stage)	RF	0.9530 ± 0.0022	0.9544 ± 0.0019	0.9555 ± 0.0016	0.9549 ± 0.0018	0.9545 ± 0.0015	0.9543 ± 0.0015	-	0.9546 ± 0.0016	0.9547 ± 0.0015	-
	NB	0.8243 ± 0.0030	0.8255 ± 0.0032	0.8371 ± 0.0040	0.8337 ± 0.0039	0.8315 ± 0.0042	0.8244 ± 0.0049	-	0.8220 ± 0.0049	0.8214 ± 0.0047	-
No Stage & Baselines	RF	0.8665 ± 0.0014	0.9005 ± 0.0014	0.9369 ± 0.0015	0.9436 ± 0.0016	0.9452 ± 0.0019	0.9468 ± 0.0015	0.9479 ± 0.0020	-	-	-
	NB	0.8185 ± 0.0019	0.7961 ± 0.0026	0.7841 ± 0.0022	0.7703 ± 0.0023	0.7655 ± 0.0024	0.7460 ± 0.0038	0.7392 ± 0.0033	-	-	-

Table D.5: 2B raw: Predicting the next hour (mean ± standard deviation)

		Number of features									
		5	10	20	30	40	50	57 (All)	60	63 (All)	64 (All)
Full Dataset	RF	0.9452 ± 0.0015	0.9491 ± 0.0016	0.9469 ± 0.0014	0.9451 ± 0.0020	0.9439 ± 0.0015	0.9433 ± 0.0016	-	0.9431 ± 0.0014	-	0.9434 ± 0.0016
	NB	0.9415 ± 0.0011	0.9092 ± 0.0028	0.8949 ± 0.0016	0.8892 ± 0.0028	0.8800 ± 0.0041	0.8696 ± 0.0053	-	0.8565 ± 0.0048	-	0.8556 ± 0.0050
Full Dataset (No Stage)	RF	0.9456 ± 0.0013	0.9473 ± 0.0017	0.9468 ± 0.0019	0.9459 ± 0.0015	0.9447 ± 0.0018	0.9444 ± 0.0015	-	0.9445 ± 0.0014	0.9441 ± 0.0015	-
	NB	0.7713 ± 0.0038	0.7680 ± 0.0040	0.7927 ± 0.0029	0.7989 ± 0.0032	0.7912 ± 0.0044	0.7883 ± 0.0047	-	0.7836 ± 0.0044	0.7833 ± 0.0041	-
No Stage & Baselines	RF	0.8604 ± 0.0012	0.8704 ± 0.0015	0.9290 ± 0.0018	0.9348 ± 0.0019	0.9364 ± 0.0017	0.9376 ± 0.0015	0.9394 ± 0.0018	-	-	-
	NB	0.8151 ± 0.0013	0.8094 ± 0.0009	0.7798 ± 0.0025	0.7673 ± 0.0030	0.7631 ± 0.0034	0.7438 ± 0.0040	0.7371 ± 0.0027	-	-	-

Table D.6: Creat: Predicting the next hour (mean ± standard deviation)

		Number of features									
		5	10	20	30	40	50	57 (All)	60	63 (All)	64 (All)
Full Dataset	RF	0.9968 ± 0.0004	0.9967 ± 0.0003	0.9966 ± 0.0004	0.9956 ± 0.0005	0.9960 ± 0.0004	0.9962 ± 0.0005	-	0.9965 ± 0.0004	-	0.9965 ± 0.0004
	NB	0.9960 ± 0.0005	0.9940 ± 0.0007	0.9917 ± 0.0010	0.9914 ± 0.0010	0.9913 ± 0.0010	0.9910 ± 0.0011	-	0.9909 ± 0.0010	-	0.9909 ± 0.0010
Full Dataset (No Stage)	RF	0.9968 ± 0.0004	0.9967 ± 0.0004	0.9963 ± 0.0005	0.9955 ± 0.0005	0.9956 ± 0.0005	0.9959 ± 0.0005	-	0.9963 ± 0.0004	0.9963 ± 0.0004	-
	NB	0.7529 ± 0.0014	0.7382 ± 0.0022	0.7316 ± 0.0035	0.7273 ± 0.0023	0.7158 ± 0.0024	0.7007 ± 0.0028	-	0.7049 ± 0.0024	0.6978 ± 0.0030	-
No Stage & Baselines	RF	0.9959 ± 0.0004	0.9966 ± 0.0004	0.9962 ± 0.0005	0.9960 ± 0.0005	0.9956 ± 0.0005	0.9961 ± 0.0004	0.9962 ± 0.0004	-	-	-
	NB	0.7666 ± 0.0012	0.7526 ± 0.0017	0.7432 ± 0.0019	0.7265 ± 0.0019	0.7065 ± 0.0024	0.7044 ± 0.0025	0.6948 ± 0.0028	-	-	-

Appendix E

Results for Reduced Patients

Table E.1: Reduced 2B: Predicting the current hour (mean ± standard deviation)

		Number of features								
		5	10	20	30	40	50	57 (All)	60	63 (All)
Full Dataset (No Stage)	RF	0.9814 ± 0.0026	0.9841 ± 0.0032	0.9844 ± 0.0033	0.9847 ± 0.0037	0.9840 ± 0.0030	0.9847 ± 0.0032	-	0.9844 ± 0.0033	0.9847 ± 0.0033
	NB	0.8550 ± 0.0092	0.8586 ± 0.0091	0.8386 ± 0.0112	0.8161 ± 0.0121	0.8010 ± 0.0116	0.7709 ± 0.0120	-	0.7550 ± 0.0146	0.7544 ± 0.0156
No Stage & Baselines	RF	0.6378 ± 0.0105	0.8202 ± 0.0087	0.8779 ± 0.0062	0.8829 ± 0.0069	0.8859 ± 0.0078	0.8870 ± 0.0089	0.8887 ± 0.0090	-	-
	NB	0.4908 ± 0.0042	0.5223 ± 0.0095	0.5066 ± 0.0108	0.4887 ± 0.0147	0.4125 ± 0.0168	0.3967 ± 0.0198	0.4065 ± 0.0167	-	-

Table E.2: Reduced 2B Raw: Predicting the current hour (mean ± standard deviation)

		Number of features								
		5	10	20	30	40	50	57 (All)	60	63 (All)
Full Dataset (No Stage)	RF	0.9730 ± 0.0036	0.9761 ± 0.0038	0.9781 ± 0.0025	0.9782 ± 0.0031	0.9787 ± 0.0029	0.9781 ± 0.0027	-	0.9781 ± 0.0028	0.9779 ± 0.0032
	NB	0.8134 ± 0.0063	0.8171 ± 0.0079	0.8151 ± 0.0069	0.7942 ± 0.0045	0.7886 ± 0.0064	0.7675 ± 0.0065	-	0.7509 ± 0.0073	0.7483 ± 0.0051
No Stage & Baselines	RF	0.6262 ± 0.0089	0.7865 ± 0.0090	0.8639 ± 0.0086	0.8677 ± 0.0084	0.8713 ± 0.0070	0.8719 ± 0.0072	0.8748 ± 0.0066	-	-
	NB	0.4991 ± 0.0025	0.4908 ± 0.0053	0.5194 ± 0.0079	0.5151 ± 0.0092	0.5088 ± 0.0105	0.4718 ± 0.0115	0.4749 ± 0.0107	-	-

Table E.3: Reduzed Creat: Predicting the current hour (mean ± standard deviation)

		Number of features								
		5	10	20	30	40	50	57 (All)	60	63 (All)
Full Dataset (No Stage)	RF	1 ± 0	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0001	-	0.9998 ± 0.0002	0.9998 ± 0.0002
	NB	0.4694 ± 0.0135	0.5003 ± 0.0108	0.4609 ± 0.0087	0.5118 ± 0.0095	0.5403 ± 0.0048	0.5453 ± 0.0.0065	-	0.5713 ± 0.0056	0.5741 ± 0.0045
No Stage & Baselines	RF	0.9998 ± 0.0002	0.9999 ± 0.0001	0.9998 ± 0.0002	0.9998 ± 0.0002	0.9997 ± 0.0005	0.9997 ± 0.0005	0.9997 ± 0.0005	-	-
	NB	0.4364 ± 0.0108	0.4677 ± 0.0094	0.4166 ± 0.0079	0.5121 ± 0.0065	0.5287 ± 0.0064	0.5349 ± 0.0071	0.5554 ± 0.0066	-	-

Table E.4: Reduced 2B: Predicting the next hour (mean ± standard deviation)

		Number of features									
		5	10	20	30	40	50	57 (All)	60	63 (All)	64 (All)
Full Dataset	RF	0.9078 ± 0.0047	0.9183 ± 0.0044	0.9135 ± 0.0043	0.9110 ± 0.0048	0.9092 ± 0.0057	0.9094 ± 0.0049	-	0.9076 ± 0.0052	-	0.9069 ± 0.0055
	NB	0.9129 ± 0.0034	0.9032 ± 0.0027	0.8780 ± 0.0085	0.8762 ± 0.0119	0.8620 ± 0.0148	0.8264 ± 0.0211	-	0.8095 ± 0.0229	-	0.8123 ± 0.0207
Full Dataset (No Stage)	RF	0.9019 ± 0.0076	0.9075 ± 0.0062	0.9091 ± 0.0062	0.9073 ± 0.0064	0.9065 ± 0.0056	0.9059 ± 0.0069	-	0.9063 ± 0.0064	0.9057 ± 0.0070	-
	NB	0.7813 ± 0.0077	0.7785 ± 0.0084	0.7877 ± 0.0072	0.7554 ± 0.0171	0.7264 ± 0.0193	0.7087 ± 0.0173	-	0.6871 ± 0.0233	0.6886 ± 0.0243	-
No Stage & Baselines	RF	0.6362 ± 0.0055	0.8116 ± 0.0074	0.8789 ± 0.0073	0.8836 ± 0.0084	0.8884 ± 0.0067	0.8882 ± 0.0074	0.8922 ± 0.0069	-	-	-
	NB	0.4878 ± 0.0057	0.5189 ± 0.0081	0.5078 ± 0.0062	0.5138 ± 0.0106	0.4488 ± 0.0234	0.4294 ± 0.0284	0.4364 ± 0.0250	-	-	-

Table E.5: Reduced 2B raw: Predicting the next hour (mean ± standard deviation)

		Number of features									
		5	10	20	30	40	50	57 (All)	60	63 (All)	64 (All)
Full Dataset	RF	0.8865 ± 0.0082	0.8974 ± 0.0082	0.8942 ± 0.0086	0.8900 ± 0.0090	0.8885 ± 0.0084	0.8866 ± 0.0079	-	0.8865 ± 0.0091	-	0.8863 ± 0.0092
	NB	0.8828 ± 0.0059	0.8725 ± 0.0071	0.8657 ± 0.0064	0.8411 ± 0.0102	0.8446 ± 0.0093	0.8184 ± 0.0122	-	0.8050 ± 0.0131	-	0.8058 ± 0.0119
Full Dataset (No Stage)	RF	0.8873 ± 0.0064	0.8949 ± 0.0067	0.8938 ± 0.0077	0.8912 ± 0.0091	0.8899 ± 0.0086	0.8890 ± 0.0098	-	0.8872 ± 0.0091	0.8887 ± 0.0090	-
	NB	0.7667 ± 0.0104	0.7618 ± 0.0132	0.7763 ± 0.0118	0.7476 ± 0.0125	0.7490 ± 0.0115	0.7194 ± 0.0144	-	0.7142 ± 0.0126	0.7123 ± 0.0115	-
No Stage & Baselines	RF	0.6305 ± 0.0073	0.7747 ± 0.0075	0.8639 ± 0.0074	0.8673 ± 0.0069	0.8709 ± 0.0059	0.8724 ± 0.0070	0.8759 ± 0.0045	-	-	-
	NB	0.4963 ± 0.0039	0.4852 ± 0.0074	0.5236 ± 0.0060	0.5118 ± 0.0103	0.4916 ± 0.0143	0.4764 ± 0.0132	0.4797 ± 0.0130	-	-	-

Table E.6: Reduced Creat: Predicting the next hour (mean ± standard deviation)

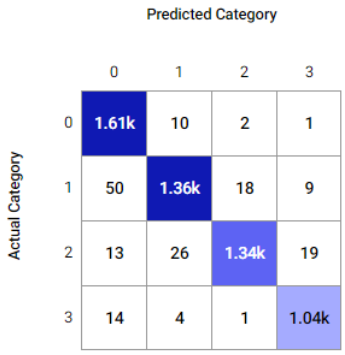
		Number of features									
		5	10	20	30	40	50	57 (All)	60	63 (All)	64 (All)
Full Dataset	RF	0.9930 ± 0.0019	0.9929 ± 0.0018	0.9926 ± 0.0018	0.9908 ± 0.0021	0.9899 ± 0.0021	0.9915 ± 0.0018	-	0.9921 ± 0.0018	-	0.9921 ± 0.0017
	NB	0.9925 ± 0.018	0.9925 ± 0.018	0.9900 ± 0.0024	0.9901 ± 0.0023	0.9899 ± 0.0023	0.9897 ± 0.0021	-	0.9894 ± 0.0022	-	0.9891 ± 0.0023
Full Dataset (No Stage)	RF	0.9930 ± 0.0018	0.9928 ± 0.0018	0.9918 ± 0.0016	0.9900 ± 0.0022	0.9900 ± 0.0021	0.9908 ± 0.0015	-	0.9917 ± 0.0018	0.9919 ± 0.0017	-
	NB	0.4665 ± 0.0104	0.4990 ± 0.0114	0.4632 ± 0.0103	0.4949 ± 0.0099	0.5284 ± 0.0107	0.5448 ± 0.0125	-	0.5632 ± 0.0108	0.5738 ± 0.0094	-
No Stage & Baselines	RF	0.9928 ± 0.0018	0.9927 ± 0.0018	0.9916 ± 0.0017	0.9912 ± 0.0016	0.9901 ± 0.0020	0.9912 ± 0.0018	0.9915 ± 0.0015	-	-	-
	NB	0.4352 ± 0.0094	0.4682 ± 0.0070	0.4469 ± 0.0085	0.5069 ± 0.0111	0.5279 ± 0.0104	0.5373 ± 0.0119	0.5547 ± 0.0098	-	-	-

Appendix F

Confusion Matrices

F.1 sCr classification system

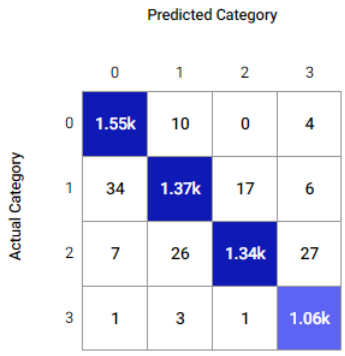
F.1.1 All features



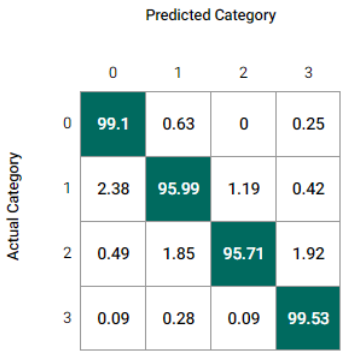
(a) Standard confusion matrix

(b) Confusion matrix with percentage by row

Figure F.1: Confusion matrix using 6h sequences

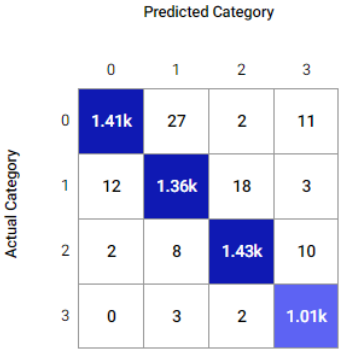


(a) Standard confusion matrix

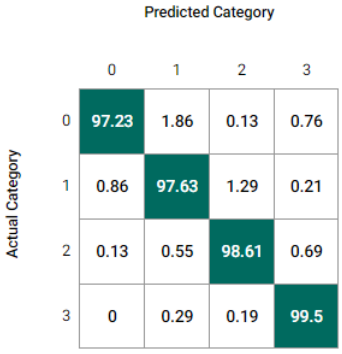


(b) Confusion matrix with percentage by row

Figure F.2: Confusion matrix using 12h sequences



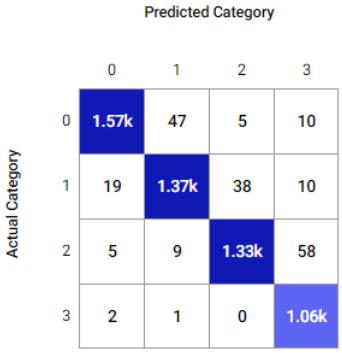
(a) Standard confusion matrix



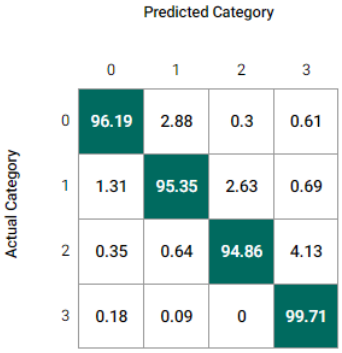
(b) Confusion matrix with percentage by row

Figure F.3: Confusion matrix using 24h sequences

F.1.2 10 most important features

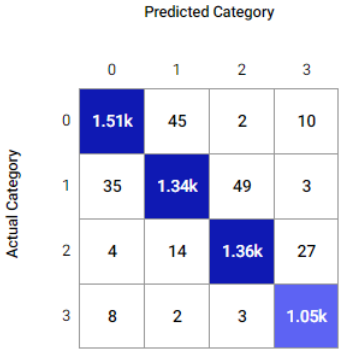


(a) Standard confusion matrix

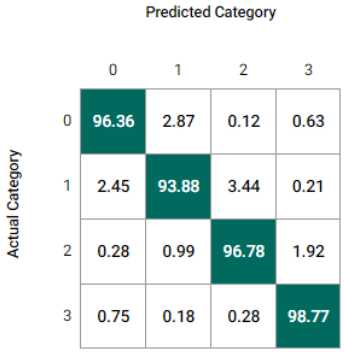


(b) Confusion matrix with percentage by row

Figure F.4: Confusion matrix using 6h sequences

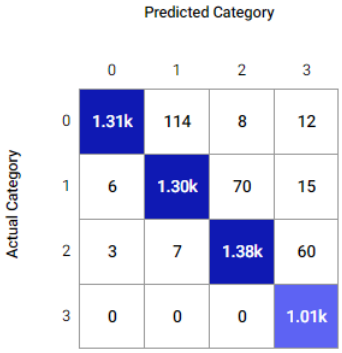


(a) Standard confusion matrix

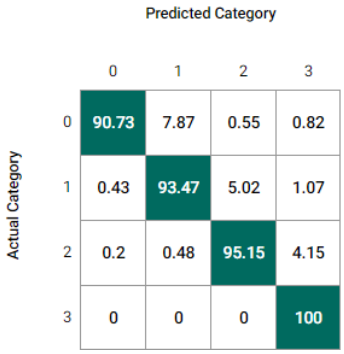


(b) Confusion matrix with percentage by row

Figure F.5: Confusion matrix using 12h sequences



(a) Standard confusion matrix

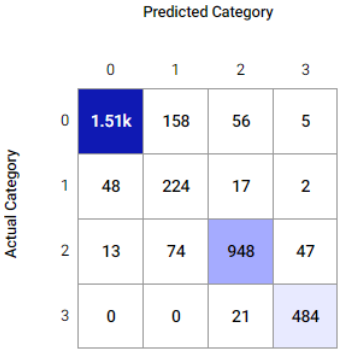


(b) Confusion matrix with percentage by row

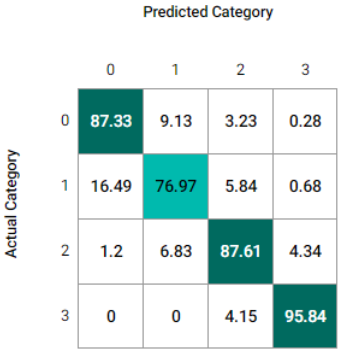
Figure F.6: Confusion matrix using 24h sequences

F.2 2B classification system

F.2.1 All features

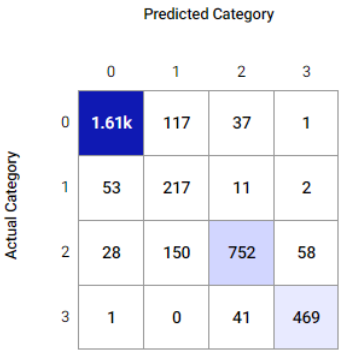


(a) Standard confusion matrix

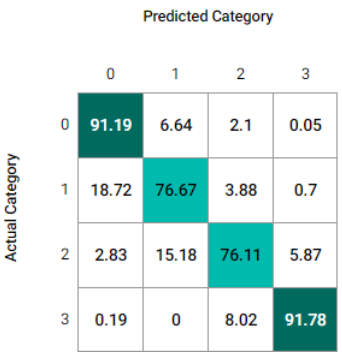


(b) Confusion matrix with percentage by row

Figure F.7: Confusion matrix using 6h sequences

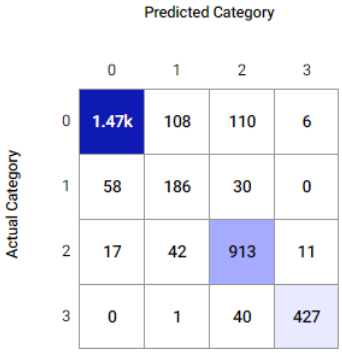


(a) Standard confusion matrix

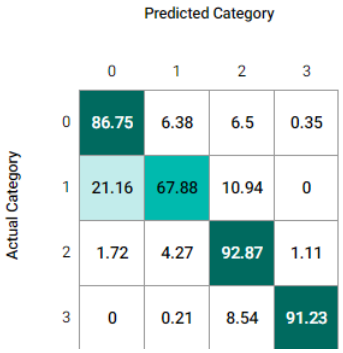


(b) Confusion matrix with percentage by row

Figure F.8: Confusion matrix using 12h sequences



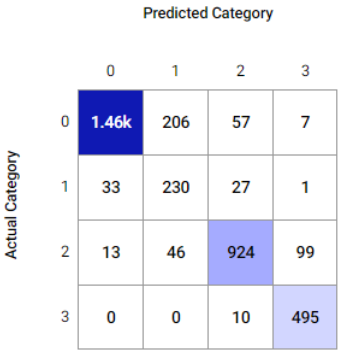
(a) Standard confusion matrix



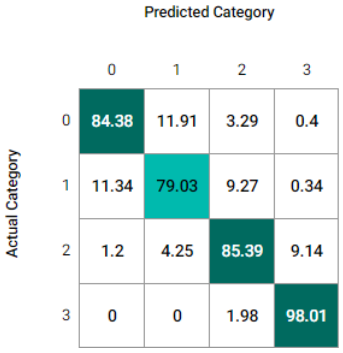
(b) Confusion matrix with percentage by row

Figure F.9: Confusion matrix using 24h sequences

F.2.2 10 most important features

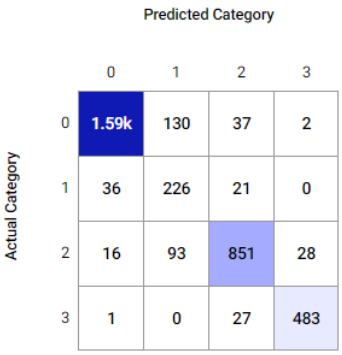


(a) Standard confusion matrix

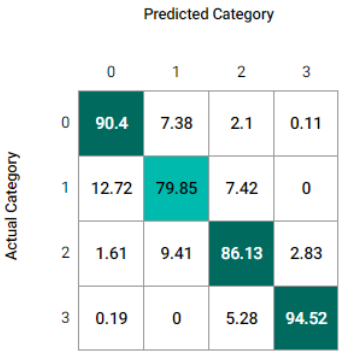


(b) Confusion matrix with percentage by row

Figure F.10: Confusion matrix using 6h sequences

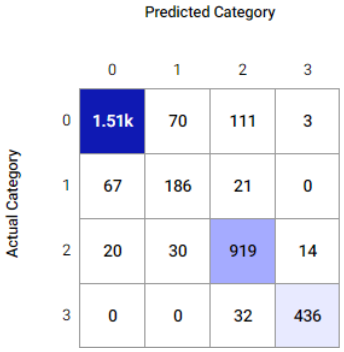


(a) Standard confusion matrix

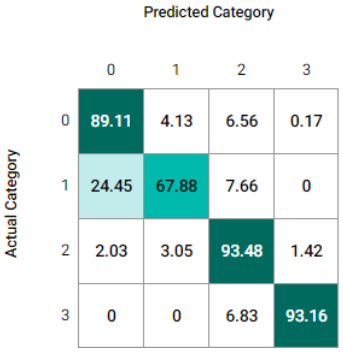


(b) Confusion matrix with percentage by row

Figure F.11: Confusion matrix using 12h sequences



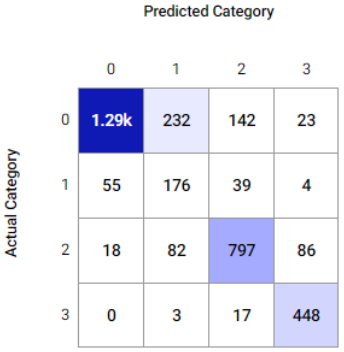
(a) Standard confusion matrix



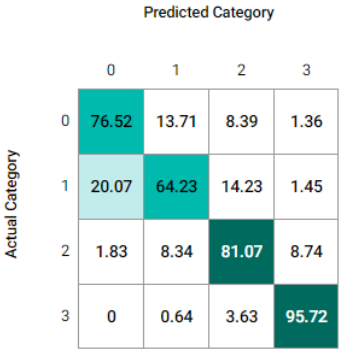
(b) Confusion matrix with percentage by row

Figure F.12: Confusion matrix using 24h sequences

F.2.3 Different learning rate

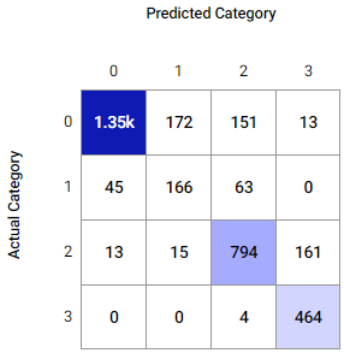


(a) Standard confusion matrix

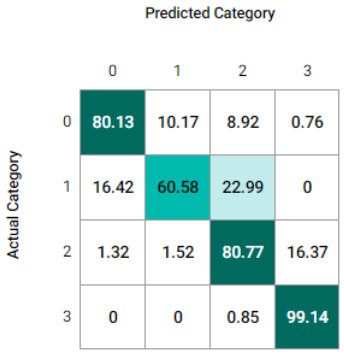


(b) Confusion matrix with percentage by row

Figure F.13: Results using a learning rate of 0.0001 with all features



(a) Standard confusion matrix

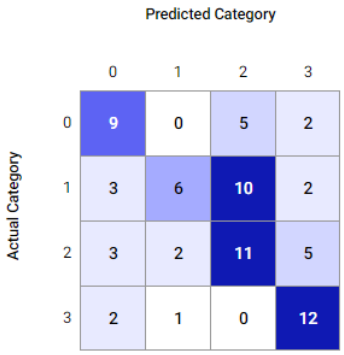


(b) Confusion matrix with percentage by row

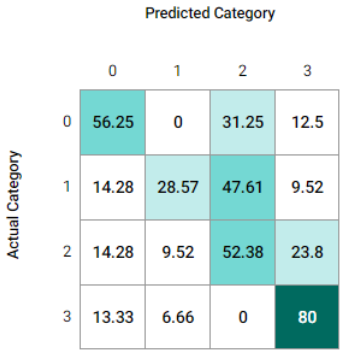
Figure F.14: Results using a learning rate of 0.0001 with 10 features

F.3 Focusing on stage alterations

F.3.1 sCr classification system



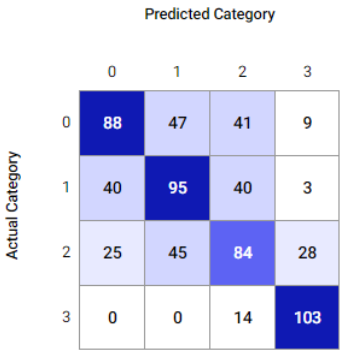
(a) Standard confusion matrix



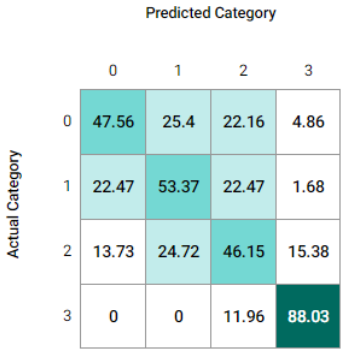
(b) Confusion matrix with percentage by row

Figure F.15: Results using the sCr classification system

F.3.2 2B classification system

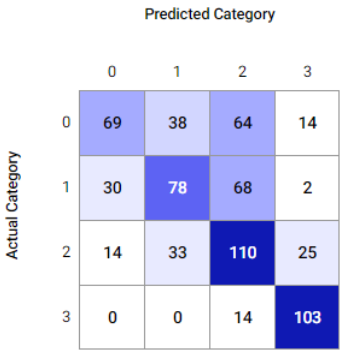


(a) Standard confusion matrix

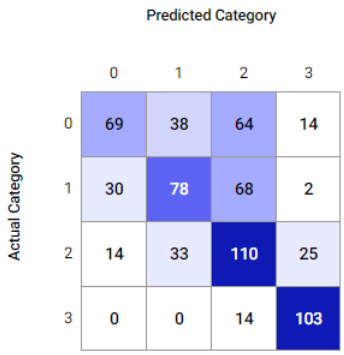


(b) Confusion matrix with percentage by row

Figure F.16: Results using the 2B classification system with learning rate of 0.00025



(a) Standard confusion matrix



(b) Confusion matrix with percentage by row

Figure F.17: Results using the 2B classification system with learning rate of 0.0005