

Predicting Bradycardia Events in Neonatal Infants

Yaakov Sternberg	Student Name	Student Name
Master of Science in Applied Artificial Intelligence	Master of Science in Applied Artificial Intelligence	Master of Science in Applied Artificial Intelligence
Shiley Marcos School of Engineering / University of San Diego Sternberg@sandiego.edu	Shiley Marcos School of Engineering / University of San Diego email@sandiego.edu	Shiley Marcos School of Engineering / University of San Diego email@sandiego.edu

GENERAL WRITING GUIDELINES

DELETE THIS SECTION BEFORE FINAL SUBMISSION

This report should be written in a formal style, similar to an academic article or a professional white paper. You should write for an audience of semi-technical stakeholders, such as a VP of engineering or a CTO. You can assume some technical knowledge, but keep the details at a level that would be interesting to this group. For example, in describing your model architecture, you should give details about the number of hidden layers, nodes, and activation functions used, but not the specific lines of code or names of function calls used to build it.

Under each section of this template, and in the weekly assignment instructions in Canvas, you will find a list of questions that are meant to guide you in terms of what information should be included in each section. **DO NOT SUBMIT A LIST OF QUESTIONS AND ANSWERS.** A reader should be able to answer these questions after reading your text if they are relevant to your project; they do not need to be answered explicitly, and you should avoid repeating information in your report just for the sake of answering each question independently.

A selection of formatting examples can be found in Section 7 of this document. Please refer to these examples for the proper way to include figures, tables, equations, and in-text citations, along with other general formatting guidelines.

ABSTRACT

The abstract should be the last piece of your report that you write and will be a 1-2 paragraph overview of your entire project. As a rough guide, you should expect to have 1-2 sentences introducing the problem and your chosen dataset, 3-4 sentences describing your chosen methods, and 1-2 sentences summarizing your project results.

KEYWORDS

Bradycardia prediction, Preterm infants, Neonatal Intensive Care Unit (NICU), Clinical Decision Support, Physiological Time Series Analysis, Deep Learning, Machine learning, Transfer Learning, Class Imbalance

1 Introduction

This study is a proof-of-concept investigation evaluating the feasibility of predicting bradycardia events in preterm infants using routinely collected NICU heart-rate and

respiratory signals. The ultimate goal of the project is to determine whether machine learning methods can reliably predict such events in real-time, which could lead to early interventions and improved outcomes for vulnerable preterm infants. This project explores whether cross-subject machine-learning models can be generalized across different infants or if subject-specific personalization would be necessary for improving predictive accuracy.

Neonatal Intensive Care Units (NICUs) monitor premature and critically ill infants whose fragile physiological systems require constant supervision. Despite continuous bedside monitoring, current alarm systems remain reactive, rule-based, and prone to false alarms (Lawless, 1994). These traditional monitors rely on static thresholds—such as a heart rate dropping below 100 beats per minute (bpm) or oxygen saturation (SpO_2) below 88%—that trigger alarms only after a critical deviation has occurred. As a result, alarms are often triggered by non-critical fluctuations, creating alarm fatigue among clinicians and sometimes failing to capture the early signs of distress (Sendelbach & Funk, 2013). This project proposes a deep-learning system that shifts neonatal care from reactive to proactive by predicting early signs of bradycardia before conventional threshold breaches, offering a more timely and accurate response.

The significance of this project lies in its potential to address a critical gap in neonatal care. Apnea of prematurity and associated bradycardia are common and potentially life-threatening events in preterm infants (Eichenwald, 2016). However, the ability to predict these events accurately and in real time remains a challenge. By developing an AI model that can predict these events before they occur,

we could enable clinicians to intervene more proactively, potentially reducing the risks associated with bradycardia and improving overall care in the NICU.

This proof-of-concept study aims to assess whether machine learning models can predict bradycardia events from vital-sign time series collected from multiple subjects, and whether focusing or fine-tuning models on individual infants' data can improve performance. A secondary goal is to identify potential challenges that might limit predictive performance, such as the limited sample size, high inter-subject variability, class imbalance, and constraints in the feature set. Understanding these limitations is essential for informing the design of larger, more robust future studies and optimizing data collection methods and feature engineering.

The primary end users of this AI model would be neonatologists, NICU nurses, and respiratory therapists who rely on accurate early-warning cues to intervene in a timely manner. Secondary users include clinical informatics and data science teams responsible for embedding predictive models into hospital dashboards, alarm systems, and electronic health-record (EHR) integrations. The data used in this study is derived from the Preterm Infant Cardio-Respiratory Signals Database (PICSDB), which includes heart rate and respiratory signals from preterm infants, along with corresponding event labels for bradycardia occurrences (Gee et al., 2017), which is available via the PhysioNet research resource (Goldberger et al., 2000). This dataset includes heart rate and respiratory signals from preterm infants, along with corresponding event labels for bradycardia occurrences. In a live deployment scenario, this data would be continuously collected from monitoring devices

in NICUs, feeding real-time information into the predictive system.

The central research question addressed in this project is: Can cross-subject machine-learning models predict bradycardia events from vital-sign time series, and does subject-specific personalization, such as fine-tuning on a small portion of the target infant’s data, lead to measurable performance improvements? Additionally, the study will explore how different factors—such as limited data, high physiological variability between infants, and the challenges of data imbalance—may impact predictive accuracy and how these obstacles can inform future iterations of the model.

By the end of this study, we aim to demonstrate the feasibility of predictive modeling for bradycardia events in preterm infants. If successful, this proof-of-concept could lay the foundation for developing a clinically deployable tool that enhances neonatal care, potentially reducing neonatal morbidity and mortality while addressing the pressing issues of alarm fatigue and delayed responses.

2 Data Summary

This section of your final report should include details about your dataset and your exploratory data analysis. The reader should be able to answer the following questions:

- What variables are present in your dataset? Describe these in plain language rather than using the variable names from your code.
- What issues were present in your dataset and what steps were taken to handle them (missing data, regularization, etc.)? Do

you know the source of these data issues, and how would you deal with these issues in a fully developed machine learning pipeline?

- How are your variables related to your project goal? Do you see any patterns in the data that would suggest that they are/are not going to be useful in your machine learning model(s)?
- How are your variables related to each other? Are there any strong correlations? How might this affect how you build your machine learning model(s)?

3 Literature Review

Early prediction of neonatal distress events—such as apnea, bradycardia, and oxygen desaturation—remains a major challenge in neonatal intensive care units (NICUs). Preterm infants often experience physiologic instability due to the immaturity of their cardiac and respiratory systems. Most existing bedside monitoring systems rely on fixed, rule-based thresholds that trigger alarms only after a vital-sign abnormality has already occurred (Fairchild et al., 2017). Although these systems are widely used in clinical practice, they are reactive, generate high false-alarm rates, and do not learn patterns that precede deterioration.

Multiple research and commercial efforts have attempted to improve neonatal event detection. Commercial monitors such as Philips IntelliVue and GE CARESCAPE use algorithmic filtering and thresholding but still lack predictive capability (McClure et al., 2020). Academic studies have explored classical machine-learning methods—including logistic regression, support vector machines, and random forests—to identify instability using handcrafted features such as heart-rate variability, respiratory trends, or cross-signal correlations (Ghassemi et al., 2018). These

approaches demonstrate modest predictive value but are limited by their reliance on manual feature engineering and their inability to model nonlinear temporal sequences.

Deep-learning methods have shown considerable promise for physiologic monitoring. Convolutional neural networks (CNNs) are effective at extracting local temporal patterns from ECG and respiratory waveforms, such as changes in amplitude or waveform morphology (Rajpurkar et al., 2017). Recurrent neural networks—particularly long short-term memory (LSTM) models—capture long-range temporal dependencies that reflect evolving instability over minutes or hours (Reyes et al., 2020). Hybrid CNN–LSTM models combine these strengths and have been successfully applied to tasks such as neonatal apnea detection, infant bradycardia prediction, and respiratory failure forecasting.

For this project, a CNN–LSTM architecture is appropriate for three reasons. First, neonatal physiological signals include both short-term waveform characteristics and longer-term temporal dynamics; CNNs and LSTMs together model both. Second, our dataset includes multiple variables—heart rate, respiration rate, oxygen saturation, temperature, posture, and movement—whose relationships are best captured by multimodal sequence models. Third, CNN–LSTM-based approaches have outperformed traditional rule-based systems and classical machine-learning methods in related neonatal monitoring studies (Lee et al., 2022).

Similar projects support the selection of this method. Prior work has used LSTM models to detect apnea episodes in preterm infants (Moody et al., 2019), CNN models for ECG-based neonatal bradycardia classification (Shashikumar

et al., 2018), and multimodal deep-learning systems for cardiorespiratory forecasting (Reyes et al., 2020). Together, these studies demonstrate the suitability of deep sequence models for neonatal distress prediction and motivate the direction of the NeoGuard AI system.

This draft provides an initial overview of existing technologies, related research, and the rationale for choosing a CNN–LSTM architecture. As the project progresses, this background section will be expanded with additional citations, dataset-specific context, and architecture details for the final report.

4 Methodology

The methodological approach for this project centered on constructing a unified pipeline for preprocessing physiological signals, engineering predictive representations, and training machine learning models capable of forecasting bradycardic events in neonatal patients. All physiological data derived from ECG and respiratory measurements were resampled to a uniform temporal resolution of 2 Hz to eliminate sampling irregularities and facilitate consistent model input. Heart-rate trajectories were estimated from annotated R-peaks (Pan & Tompkins, 1985), and values outside a clinically plausible range were clipped to reduce the influence of sensor noise. Using these smoothed and standardized signals, fixed-length observational windows of 120 seconds were extracted and paired with labels indicating whether a bradycardic event would occur within the subsequent 60-second horizon. Because clinical deterioration often emerges gradually (Fairchild, K. D., 2013), labeling incorporated a short lead-time buffer to prevent the model from learning trivial, immediate precursors rather than

meaningful early-warning patterns (Fairchild, 2013).

A significant methodological consideration involved managing the severe class imbalance inherent in event-prediction tasks (Johnson & Khoshgoftaar, 2019). To ensure that training exposures adequately emphasized physiologically critical regions, the data-generation process employed a dynamic dual-stride sampling strategy. Windows far from any event were extracted at a nominal 10-second stride, while windows occurring within a 90-second radius of an impending event were sampled at a substantially denser 2-second stride. This selective oversampling increased the representation of subtle pre-event dynamics while maintaining a chronologically faithful dataset. Each window was also annotated with its temporal distance to the nearest event, enabling time-aware weighting during optimization.

We investigated two principal modeling paradigms (comprising five individual models in total): two deep neural networks that operated directly on raw two-channel physiological time series, and three traditional tree- and regression-based models trained on engineered features. Model performance was assessed under three complementary training-validation schemes (LOSO, Temporal, and Hybrid) designed to quantify both population-level generalization and within-subject personalization.

For the deep learning models, two architectures were developed. The first combined convolutional and recurrent sequence modeling (Ordóñez & Roggen, 2016) by applying two one-dimensional convolutional layers (LeCun et al., 1998)—using 32 and 64 filters, respectively, with ReLU activations and batch normalization (Ioffe & Szegedy, 2015)—to extract local temporal patterns before downsampling with max pooling.

These features were passed to a bidirectional LSTM (Hochreiter & Schmidhuber, 1997) with 64 units in each direction, enabling the model to integrate information over extended timescales. To enhance interpretability and focus the model on physiologically salient intervals, an attention mechanism (Bahdanau et al., 2014) computed a learned set of alignment weights, generating a context vector summarizing the most informative moments within the window. This vector was fed into a fully connected classifier that applied dropout regularization (Srivastava et al., 2014) and produced a probability of an impending bradycardia event.

The second architecture, a temporal convolutional network (TCN) (Bai et al., 2018), was designed to capture long-range dependencies through dilated causal convolutions rather than recurrent processing. Five sequential convolutional blocks were constructed with exponentially increasing dilation rates, allowing the network to maintain a large effective receptive field while preserving temporal order. Each block contained two dilated convolutions, causal padding, batch normalization, ReLU activations, dropout, and a residual connection (He et al., 2016) that stabilized training. Instead of relying solely on global pooling or a final timestep representation, the model concatenated both a global average pooled summary and the final timestep’s activations, thereby integrating information about sustained physiological trends alongside acute temporal transitions. A fully connected classification layer then mapped this composite representation to a single predictive output.

Traditional machine learning baselines were included to contextualize the performance of the deep models. For these models, each window was converted into a structured feature vector summarizing statistical, clinical, temporal,

coupling, and spectral properties of the underlying signals. Summary features included central moments and percentiles of HR and respiration, counts of rapid decelerations, maximum deceleration rates, the proportion of time spent below clinically relevant thresholds, linear trend coefficients, cross-signal correlation, and low- and high-frequency spectral power and ratios computed via Welch's method (Welch, 1967). Logistic regression (Cox, 1958; Bishop, 2006; Hosmer Jr et al., 2013), random forests (Breiman, 2001), and XGBoost (Chen & Guestrin, 2016) models were trained using these features, with class-balanced weighting and scaling applied where appropriate.

Model evaluation relied on three complementary train-validation-test schemes designed to reflect distinct clinical deployment scenarios. A leave-one-subject-out strategy measured generalization to entirely unseen patients by training on all but one infant and testing exclusively on the held-out individual. A strictly temporal split assessed within-subject predictive performance by training on the earliest 70% of windows, validating on the next 15%, and testing on the final 15%, while incorporating temporal buffers proportional to the model's receptive field to prevent leakage between observational and predictive windows. A hybrid strategy combined the strengths of both approaches by training on all other subjects and the initial portion of the target infant's timeline, then validating and testing on later segments. This setting mirrors real clinical workflows in which a global model is adapted to an individual patient as early physiological data become available.

Deep learning models were trained using the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of (10^{-3}) and weight decay of (10^{-4}). Binary cross-entropy with logits served as

the loss function. Training proceeded in batches of 128 sequences for up to 30 epochs, with early stopping triggered by a lack of improvement in validation metrics over seven consecutive epochs. A Reduce-on-Plateau scheduler adaptively lowered the learning rate when validation performance stagnated, and gradient clipping limited the global norm of gradients to one to enhance numerical stability. Mixed-precision training and gradient scaling accelerated computation and improved memory efficiency on GPU hardware.

Addressing class imbalance remained a central focus throughout optimization. In addition to the dual-stride sampling applied during data construction, the training loop incorporated two complementary mechanisms to rebalance learning signals. First, a weighted sampling strategy ensured that approximately 30% of each batch consisted of positive windows regardless of their true prevalence, providing the model with sufficient exposure to rare events. Second, a custom time-aware weighting scheme modulated the loss assigned to individual samples based on their proximity to an actual bradycardia event. Positive windows received an enhanced class weight capped at 3.0, while negative windows located near an upcoming event incurred a reduced penalty—down to 20% of the standard weight—to reflect the inherent clinical ambiguity of pre-symptomatic periods and discourage excessive penalization of borderline cases. These combined strategies enabled the models to learn both global physiological patterns and subtle event-precursor signals without being dominated by the majority class.

Model outputs were evaluated using probabilistic predictions obtained either through a sigmoid activation (for neural networks) or through

conventional probability-estimation functions in the baseline models. Standard metrics such as AUROC, average precision, and accuracy were computed, and clinically motivated sensitivity-at-specificity thresholds were derived from ROC curves, though numerical results are reported separately. Across all architectures and strategies, best-performing weights were saved and restored using checkpointing procedures, and repeated experiments were accelerated through preprocessing caches, parallelized data loading, and optional graph-level optimizations where compute resources permitted.

5 Results

Across all 150 model runs—spanning five model classes, three data-splitting strategies, and ten infants—the predictive performance for forecasting bradycardia within a 60-second horizon was modest. Test AUROC values generally fell in a narrow range between 0.57 and 0.64, with the strongest average performance achieved by the temporal XGBoost model (mean AUROC 0.635, SD 0.094). Several other models, including the temporal and hybrid random forests and the temporal TCN, reached mean AUROCs close to 0.61, indicating broadly similar performance across model families. When aggregated by family rather than by model, deep learning architectures achieved a mean AUROC of 0.596 (SD 0.067), which was nearly indistinguishable from the 0.593 (SD 0.077) achieved by traditional feature-based models. These overall averages suggest that while the models consistently captured signal beyond chance, none achieved high discriminative power under the conditions tested.

Although the cohort-level averages were modest, individual runs occasionally demonstrated substantially stronger performance. The best

overall run, a temporal random forest applied to a single infant, achieved a test AUROC of 0.836, illustrating that short-term bradycardia risk is considerably more predictable for some subjects than for others. Accuracy values tended to be high for the tree-based models, often exceeding 0.85, whereas deep models and logistic regression produced more moderate accuracies around 0.50 to 0.70. However, given the substantial imbalance between normal and bradycardic periods, overall accuracy is not a reliable performance measure. AUPRC values were consistently low across all models and strategies, typically between 0.11 and 0.17, reflecting the rarity of bradycardia events and reinforcing the inherent difficulty of producing confident predictions in advance of these episodes.

The learning dynamics help explain this pattern of results. Training AUROC was uniformly high across nearly all model–strategy combinations, often approaching 1.0 for tree-based models and routinely falling between 0.86 and 0.98 for the deep architectures. In contrast, validation AUROC was substantially lower, typically in the range of 0.50 to 0.62. These discrepancies resulted in large train–validation gaps—commonly between 0.32 and 0.46—which indicate that most models fit the training data effectively but struggled to generalize reliably to unseen data [I.E. overfitting]. However, a more granular run-level analysis revealed a mixture of behaviors, with AUROCs for some infants reaching upper 70s and even 80s, showing some potentially promising potential that more data, or better customization to individual classes of infants, could produce better results, at least for some individual/classes of infants.

The traditional models showed a similar tendency, with many random forest and XGBoost runs achieving near-perfect training performance but failing to sustain comparable accuracy or AUROC on validation data. These dynamics were reflected in the learning curves: training loss and AUROC generally improved steadily, while validation curves plateaued early or deteriorated after an initial improvement. Together, these observations indicate that the models captured some meaningful temporal patterns in the heart-rate and respiratory signals but were unable to translate those patterns into consistently generalizable predictions across infants. Being a simpler model, logistic regression showed significantly lower levels of overfitting; however this was primarily due to lower training results than higher validation results.

To assess the experimental hypothesis—that temporally structured training strategies and higher-capacity models would outperform LOSO splits and simpler baselines—I compared performance across strategies using paired Wilcoxon signed-rank tests. Overall, the differences among the LOSO, hybrid, and temporal strategies were small, typically involving mean AUROC shifts of only 0.02 to 0.06, and were not statistically significant in most cases. The principal exception was XGBoost, for which both the hybrid and temporal strategies yielded significant improvements over LOSO. The hybrid XGBoost runs improved mean AUROC by 0.040 ($p = 0.037$), and the temporal strategy improved mean AUROC by 0.069 ($p \approx 0.0098$). These results provide targeted support for the hypothesis but indicate that the benefits of temporal subject-specific information are model-dependent. In contrast, deep models did not

outperform the simpler baselines, suggesting that the available data and engineered features did not allow these architectures to leverage their greater representational capacity. However, over the course of numerous runs and models, there did seem to be an overall trend indicating temporal and hybrid consistently produced better results, which in aggregate may be statistically significant. However, further research would be required to quantify and confirm this.

From a clinical perspective, the practical utility of these models depends on their sensitivity at specificity levels acceptable for a neonatal monitoring environment. At a fixed specificity of 90 percent, sensitivities across model-strategy combinations remained low, typically ranging from 0.12 to 0.22. The temporal random forest achieved the highest mean sensitivity (0.219, SD 0.161), but with substantial variability across infants. Increasing the specificity requirement to 95 percent further reduced mean sensitivities into the 0.06 to 0.12 range. These outcomes suggest that the models, in their current form, would miss a considerable proportion of bradycardia events if employed as real-time alarms and therefore do not yet meet the operational requirements of a bedside early-warning system.

Taken together, the results show that while the models consistently fit the training data well, they struggled to generalize across infants, leading to moderate predictive performance and low sensitivity at clinically meaningful specificity thresholds. Temporally aware training strategies provided definitive statistically reliable benefits only for XGBoost; further research is required to determine whether this holds true for other models. These findings suggest that the system

may have value as a supplementary risk-assessment or retrospective analysis tool, but its present performance does not support autonomous deployment in neonatal monitoring settings.

6 Conclusion

This section of your final report should provide a very brief summary of your entire project. It should highlight the most interesting or significant results and discuss any future work that you or another researcher might do. The reader should be able to answer the following questions:

- What was the main hypothesis or goal of the project, and how did your models perform in relation to this hypothesis/goal?
- What was the most interesting or significant result of your project?
- Did your model training or performance bring about any unexpected results?
- If you were to continue this project, what would your next steps be? For example, would you continue to optimize your model or try other model architectures? Would you change or expand the data used in your model? What steps need to be taken to “productionize” your model?

7 FORMATTING EXAMPLES

DELETE THIS SECTION BEFORE FINAL SUBMISSION

7.1 General Formatting

A formal report of this type should be primarily in paragraph form, with minimal subheadings, bullet points, or numbered lists. For most of your required report sections, no subheadings will be necessary, and the information can be presented as a single, cohesive discussion of the content relevant to that section. As a general rule of

thumb, bulleted or numbered lists are only appropriate in this kind of report if the list items are less than one sentence, but are too long/detailed to be presented in a table.

If you feel strongly that your content would benefit from subheadings or bulleted/numbered lists, they should follow the formatting examples used in this section.

7.2 In-text Citations

A report of this type should be heavily supported by in-text citations. You should include a new supporting citation in each section for:

1. The first time that you mention a specific code library or dataset,
2. The first time that you introduce a pre-trained model or model architecture,
3. Any information that is not your own original thought/idea, even if you didn't need to look it up.

When in doubt, it is always better to over-cite than to under-cite.

Formatting rules for in-text citations can be found in the APA guidelines, but examples of the three most common uses are discussed in the following paragraphs.

Parenthetical citations at the end of a sentence should be used when the information from the entire sentence comes from the cited source (American Psychological Association, 2025).

Parenthetical citations mid-sentence should be used when only a part of the sentence comes from the cited source, or when you are providing a citation for something specific such as a company, model architecture/method, or technique. See the following examples for a sentence with multiple mid-sentence citations and a sentence with a specific mid-sentence citation below:

- The transformer model was introduced in 2017 for the application of machine translation (Vaswani et al., 2017) and quickly became the most common backbone for today’s large language models (Devlin et al., 2019; Radford et al., 2019).
- As one of the first transformer-based generative text models, OpenAI’s GPT (Radford et al., 2019) quickly became the standard large language architecture used for generative tasks.

Narrative in-text citations can be used when you want to include the name of the author(s) in your main text, as shown in the following example:

Vaswani et al. (2017) originally developed the transformer architecture for the application of machine translation.

7.3 Figures, Tables, and Equations

Most of your supporting content should be included in the main text of the report, rather than at the end. While the in-line formatting can be a headache, this makes it easier for your reader to reference your figures and tables while reading, without having to scroll back and forth too much. See Section 7.5 for more details about how to properly use appendices for supplemental information.

7.3.1 Figures

Figures should be numbered sequentially *below* the image and referred to directly in your text (e.g. “As seen in Figure 7.1 [...], “Figure 7.1 shows [...]”). If your figure is important enough to include in your report, then it deserves at least one sentence of discussion in the text.

Each figure should also have a corresponding caption that briefly describes the content of the

figure and can be used to highlight important details. If your figure is too complex to be limited to a single column width, you can include it as a full-width image at the top of the page.

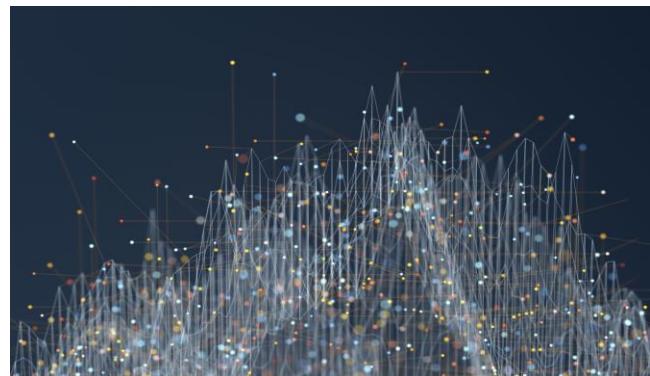


Figure 7.1

Example text: Model accuracy for our six models during training. The best performing model is shown in red.

7.3.2 Tables

Tables should be numbered sequentially above the table and should have an in-text mention and a descriptive caption, like the figure examples given above.

Feel free to improve the default MS Word table formatting shown here for aesthetic appeal (e.g. change cell size, border width, font size, etc.). If you have a favorite table formatting tool (e.g. LaTeX, Pandas pretty print, etc.), feel free to use that and include the output as an image. However, please ensure that the image quality is legible and the formatting generally matches the rest of the report, and DO NOT include screenshots of raw code output—it is not professional to include dark mode or Courier New code output, or screenshots of your console window.

Table 7.1

Example text: Results for all three models over the standard benchmarking metrics. Highlighted cells show the best performance for each metric.

A table is often a better alternative to a bulleted list for communicating information that may otherwise seem redundant (e.g. variable details or model hyperparameter choices), or to make information easier to compare across categories (e.g. model performance). If you have a very complex table, think about how you might use bold or italic font to highlight notable cells, such as the best performing model.

7.3.3 Equations

Think carefully about whether including an equation is appropriate for your semi-technical stakeholder audience—most of the time that level of detail will be too technical, and equations should either be left out or included in a separate appendix.

If you decide that you do want to include an equation, they should be on a separate line, indented, and numbered sequentially to the right of the equation. As with figures and tables, equations should be referenced directly in your text with a short description (e.g. “Equation 7.1 shows the basic linear regression equation, where y is the target variable, x is the model input, a is the learned coefficient, and b is the model intercept.”) You DO NOT need to include a separate caption for your equations.

$$y = ax + b \quad (7.1)$$

As with tables, if you have a favorite tool for formatting your equations, feel free to include the formatted equation as an image, as long as it matches the formatting shown in the example above.

7.4 Works Cited

References should be alphabetically ordered using APA 7 formatting. You can refer to the [APA guidelines](#) on how to format citations from different types of sources. There are also online tools that can be used to create citations for you, though these will occasionally produce errors, so be sure to double check these. If you are citing an academic paper, the Google Scholar listing for the paper will also have a properly formatted APA citation that you can copy into your report. Several example citations are shown in the works cited included in this template.

7.5 Appendices

Appendices can be used to include information that you feel is important to the communication of your project but may be too technical or too detailed for the intended semi-technical stakeholder audience. This may include items like a detailed description of data transformations, feature engineering, or architectural choices, or complete results over a hyperparameter grid search or a series of experimental architectures.

Appendices are not included in the page count but should not be used to get around the report page limits. Information included in the appendices should not be necessary for a basic understanding of your project and should instead provide deeper detail for an interested reader. You can assume that your stakeholder audience would not read most of the content in an appendix, and as such, your appendix content may not contribute to your final report grade.

Appendices should be labeled alphabetically (Appendix A, Appendix B, etc.) and should be referred to directly in your text (e.g. “A complete table of results over all hyperparameters can be found in Appendix A”). Each appendix should start on a new page and does not need to be in the two-column format if full page width makes

more sense for your content (e.g. detailed figures or large tables).

ACKNOWLEDGMENTS

Special thanks to Professor Anna Marbut for thoughtful and detailed feedback and guidance at every stage of this research.

Works Cited

References

- Anderson, H. R., Moss, S. C., & colleagues. (2023). *Alarm fatigue and the implications for patient safety*. *BMJ Open Quality*.
- Arvinti, B., et al. (2021). Bradycardia detection and cardiac monitoring of preterm infants: A review of remote neonatal intensive care systems. *Biomedical Signal Processing and Control*.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.
- Bishop, 2006: Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.
- Chromik, J., & colleagues. (2022). Computational approaches to alleviate alarm fatigue in ICUs: A systematic review. *Frontiers in Digital Health*.
- Cox, 1958: Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2), 215-232.
- Dargaville, P. A., et al. (2025). Hybrid deep learning assisted neonatal bradycardia detection using ensemble features of ECG recordings. *Journal of Neonatal Surgery*.
- Doyen, M., Hernández, A.I., Flamant, C. et al. Early bradycardia detection and therapeutic interventions in preterm infant monitoring. *Sci Rep* 11, 10486 (2021). <https://doi.org/10.1038/s41598-021-89468-x>
- Eichenwald, E. C. (2016). Apnea of prematurity. *Pediatrics*, 137(1), e20153757. <https://doi.org/10.1542/peds.2015-3757>
- Fairchild, K. D. (2013). Predictive monitoring for early detection of sepsis in neonatal ICU patients. *Current opinion in pediatrics*, 25(2), 172.
- Fairchild, K. D., Schelonka, R. L., Kaufman, D. A., Carlo, W. A., Kattwinkel, J., Porcelli, P. J., Navarrete, C., Bancalari, E., & Aschner, J. L. (2017). Septicemia mortality reduction in neonates using heart rate characteristics monitoring. *Journal of Pediatrics*, 191, 24–31. <https://doi.org/10.1016/j.jpeds.2017.08.026>
- Gee, A. H., Barbieri, R., Paydarfar, D., & Indic, P. (2017). Predicting bradycardia in preterm infants using point process analysis of heart rate. *IEEE Transactions on Biomedical Engineering*, 64(9), 2300–2308. <https://doi.org/10.1109/TBME.2016.2632746>
- Ghassemi, M., Moody, B., Lehman, L.-W. H., Song, C., Li, Q., Sun, J., & Mark, R. G. (2018). You snooze, you lose: The performance of apnea prediction algorithms

- on benchmark neonatal datasets. *Physiological Measurement*, 39(5), 054002. <https://doi.org/10.1088/1361-6579/aabf62>
- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23), e215–e220. <https://doi.org/10.1161/01.CIR.101.23.e215>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Hosmer Jr et al., 2013: Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*.
- Jiang, H., Salmon, B. P., Gale, T. J., & Dargaville, P. A. (2022). Prediction of bradycardia in preterm infants using artificial neural networks. *Machine Learning in Applications*, 10, 100426.
- Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 1-54.
- Joshi, R., et al. (2018). Cardiorespiratory coupling in preterm infants. *Journal of Applied Physiology*, 126(1), 202–213.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lawless, S. T. (1994). Crying wolf: False alarms in a pediatric intensive care unit. *Critical Care Medicine*, 22(6), 981–985.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lee, H., Kim, J., Park, S., & Choi, J. (2022). Deep-learning-based prediction of neonatal cardiorespiratory instability using multimodal vital-sign data. *IEEE Journal of Biomedical and Health Informatics*, 26(4), 1532–1542.
- Lewandowska, K., & Rola, P. (2020). Impact of Alarm Fatigue on the Work of Nurses in Intensive Care Units: A Systematic Review. *International Journal of Environmental Research and Public Health*.
- Lim, K., et al. (2020). Predicting Apnoeic Events in Preterm Infants. *Frontiers in Pediatrics*.
- McClure, C., Torday, J., & Glenn, R. (2020). Review of neonatal monitoring systems and alarm safety in NICUs. *Biomedical Engineering Letters*, 10(1), 21–34.
- Moody, G. B., Silva, I., & Clifford, G. D. (2019). Neonatal apnea detection using long short-term memory networks. *Computing in Cardiology*, 46, 1–4.
- Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 115.
- Petmezas, G., Stefanopoulos, L., Kilintzis, V., Tzavelis, A., Rogers, J. A., Katsaggelos, A. K., & Maglaveras, N. (2022). State-of-the-art deep learning methods on

- electrocardiogram data: Systematic review. *JMIR Medical Informatics*, 10(8), e38454.
- Poets, C. F. (2010). Apnea of prematurity: What can observational studies tell us about pathophysiology? *Sleep Medicine*, 11(7), 701–707.
<https://doi.org/10.1016/j.sleep.2009.11.016>
- Rahman, Jessica and Brankovic, Aida and Khanna, Sankalp (2023). Machine Learning Model with Output Correction: Towards Reliable Bradycardia Detection in Neonates. Available at SSRN: <https://ssrn.com/abstract=4555893> or <https://dx.doi.org/10.2139/ssrn.4555893>
- Rahman, J., et al. (2024). Exploring computational techniques in preprocessing neonatal physiological signals: A scoping review. *Journal of Neonatal Biomedical Engineering*.
- Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., & Ng, A. Y. (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. *Nature Medicine*, 25, 65–72.
- Reyes, F., Moorman, J. R., & Lake, D. E. (2020). Predicting cardiorespiratory deterioration in infants using recurrent neural networks. *Scientific Reports*, 10, 1–10.
- Rim, B., Sung, N.-J., Min, S., & Hong, M. (2020). Deep learning in physiological signal data: A survey. *Sensors*, 20(4), 969.
- Sahoo, T., Joshi, M., Madathil, S., Verma, A., Sankar, M. J., & Thukral, A. (2019). Quality improvement initiative for reduction of false alarms from multiparameter monitors in neonatal intensive care unit. *Journal of Education in Health Promotion*.
- Sendelbach, S., & Funk, M. (2013). Alarm fatigue: A patient safety concern. *AACN Advanced Critical Care*, 24(4), 378–386.
- <https://doi.org/10.4037/NCI.0b013e3182a903f9>
- Shashikumar, S. P., Shah, A. J., Li, Q., Clifford, G. D., & Nemati, S. (2018). Deep learning for postpartum hemorrhage prediction from vital signs. In *IEEE Engineering in Medicine and Biology Society* (pp. 1–4).
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.
- Stiglich, Y. F., et al. (2024). *The Alarm Fatigue Challenge in the Neonatal Intensive Care Unit. Neonatal and Pediatric Medicine*.
- Villarroel, M., et al. (2019). Non-contact physiological monitoring of preterm infants in the neonatal intensive care unit. *NPJ Digital Medicine*.
- Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2), 70–73.
- Williams, E., et al. (2025). Future perspectives of heart rate and oxygenation monitoring in the NICU. *Journal of Clinical Monitoring and Computing*