



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

PREDICCIÓN DE CURVAS DE LUZ DE SUPERNOVAS MEDIANTE REDES  
NEURONALES RECURRENTES

TESIS PARA OPTAR AL GRADO DE  
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

PABLO ANDRÉS MONTERO SALVATIERRA

PROFESOR GUÍA:  
PABLO ESTÉVEZ VALENCIA

MIEMBROS DE LA COMISIÓN:  
FRANCISCO FÖRSTER BURÓN  
MARCOS ORCHARD CONCHA

Este trabajo ha sido parcialmente financiado por ANID, Iniciativa Científica Milenio,  
ICN12009 y Fondecyt 1220829.

SANTIAGO DE CHILE  
2023

RESUMEN DE LA TESIS PARA OPTAR AL  
GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA  
POR: PABLO ANDRÉS MONTERO SALVATIERRA  
FECHA: 2023  
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

## PREDICCIÓN DE CURVAS DE LUZ DE SUPERNOVAS MEDIANTE REDES NEURONALES RECURRENTES

La astronomía es una ciencia que estudia el universo. Para entenderlo, los astrónomos analizan astros y eventos astronómicos. Un evento de interés corresponde a la explosión de una estrella, la supernova. Este es un evento transiente caracterizado por su rápido incremento de brillo y posterior decaimiento. Esto se estudia midiendo el brillo en el tiempo, lo que se conoce como curva de luz. Las curvas de las supernovas tienen características que dificultan su análisis: son escasas, irregulares, multivariadas, poseen largo variable, etc. En esta tesis se propone un predictor de curvas de luz de supernovas basado en redes neuronales recurrentes, denominado Supernovae Forecaster (SNeF). Además, se proponen estrategias, basadas en el error de observación y un modelo paramétrico para mejorar el entrenamiento, la predicción del modelo, y evaluar su desempeño. Este trabajo es pionero en la predicción de curvas de supernovas. El desempeño del modelo se evalúa en la base de datos ALeRCE-ZTF. El SNeF obtiene 0.348 y 7.479 en las métricas MASE y sMAPE respectivamente para las supernovas de tipo Ia, las que superan al modelo propuesto por el *broker* ALeRCE (0.629 y 13.846 respectivamente). Finalmente, dado un intervalo de predicción del 95%, el 98% de las observaciones fueron capturadas.

*Para mi madre, abuela y Mini. Muchas gracias por todo.*

# Agradecimientos

En primer lugar agradezco a mi madre, por todos los valores y enseñanzas que me entregó a lo largo de mi vida. Gracias a ella aprendí la importancia de la paciencia, la perseverancia y el optimismo. Aprendí que cualquier desafío, por más duro que sea, puede ser enfrentado con una gran sonrisa. Gracias por darme las herramientas que me han permitido crecer personal y profesionalmente. Sé que aun que ya no estés conmigo, me seguirás acompañando y guiando como siempre lo hiciste. Por todo esto y mucho más, muchas gracias.

Agradezco a mi esposa, Daniela por acompañarme y brindarme todo el apoyo tanto en el desarrollo de este trabajo como en la vida. Has sido un pilar muy importante para poder sacar adelante este trabajo. Gracias por todos los buenos momentos, la alegría y tu eterna compañía.

Agradezco también el enorme apoyo de mi familia, por la gran confianza y constante apoyo que he recibido durante toda mi vida. Sin duda han sido parte importante en mi formación. Finalmente agradezco a mi abuela por ser un pilar fundamental en los primeros años de mi vida y que a pesar que ya no esté conmigo, aún sigo sintiendo su apoyo.

Agradezco a todos mis amigos por siempre estar ahí, por los buenos momentos. Hago una mención especial al Óscar y al Joaquín por su constante apañe y valiosas discusiones. Agradezco a la Francisca por apoyarme en el último periodo de esta tesis. Agradezco a los FC por el apoyo entregado. Agradezco a las Ñañas por todos los buenos momentos vividos, los taca taca, los almuerzo, las trasnochadas, entre otras.

Agradezco a mis gatos por acompañarme en las largas jornadas de estudios y de trasnochadas. Mención honrosa a la Mini por acompañarme en más de la mitad de mi vida y a la Ramita por sus sabios consejos.

De igual forma agradezco a mi profesor guía Pablo Estévez por su gran apoyo, guía y comprensión durante el desarrollo de mi tesis. Agradezco a mi equipo de Analytics Personas también por el apoyo y las valiosas discusiones.

Este trabajo fue parcialmente financiado por el Instituto Milenio de Astrofísica IC12009 y el proyecto FONDECYT - 1220829.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Trabajos astronómicos relacionados . . . . .	2
1.3. Definición del problema . . . . .	4
1.4. Hipótesis . . . . .	4
1.5. Objetivos . . . . .	4
1.5.1. Objetivo general . . . . .	4
1.5.2. Objetivos específicos . . . . .	5
1.6. Principales contribuciones . . . . .	5
1.7. Estructura de la Tesis . . . . .	5
1.8. Acrónimos . . . . .	6
<b>2. Antecedentes</b>	<b>8</b>
2.1. Astronomía . . . . .	8
2.1.1. Magnitud aparente . . . . .	8
2.1.2. Flujo fotométrico . . . . .	8
2.1.3. Transformación de magnitud a flujo . . . . .	9
2.1.4. Supernovas . . . . .	9
2.1.5. Modelo paramétrico de supernovas . . . . .	9
2.1.6. Curvas de luz . . . . .	11
2.1.7. Sondeos, conjuntos de datos y <i>brokers</i> . . . . .	11
2.2. Series de tiempo . . . . .	13
2.2.1. Series continuas o discretas . . . . .	13
2.2.2. Series univariadas y multivariadas . . . . .	14
2.2.3. Series regularmente e irregularmente muestreadas . . . . .	14
2.2.4. Series con valores faltantes . . . . .	15
2.3. Aprendizaje profundo . . . . .	15

2.3.1.	Redes neuronales artificiales . . . . .	15
2.3.2.	Redes neuronales recurrentes . . . . .	16
2.3.2.1.	Vanilla RNNs . . . . .	16
2.3.2.2.	Long Short-Term Memory . . . . .	17
2.3.2.3.	Gated Recurrent Unit (GRU) . . . . .	17
2.3.3.	Entrenamiento de redes neuronales . . . . .	18
2.3.3.1.	Separación de base de datos . . . . .	19
2.3.4.	Técnicas de regularización . . . . .	20
2.3.4.1.	Detención temprana . . . . .	20
2.3.4.2.	Perturbar datos de entrada . . . . .	20
2.3.4.3.	Dropout . . . . .	20
2.3.4.4.	Linear Warm-Up Schedule . . . . .	21
2.3.4.5.	Normalización por capas . . . . .	21
2.4.	Aprendizaje profundo en tareas de predicción . . . . .	21
2.4.1.	Predicciones . . . . .	21
2.4.1.1.	Formalización del problema . . . . .	22
2.4.1.2.	Modelos de predicción puntual . . . . .	22
2.4.1.3.	Modelos de predicción probabilística . . . . .	23
2.4.2.	Predicción de series de tiempo con RNNs . . . . .	23
2.4.2.1.	Características o covariables . . . . .	24
2.4.3.	Irregularidad y pérdida de valores . . . . .	25
2.4.3.1.	Imputación . . . . .	25
2.4.3.2.	Modelos . . . . .	26
2.5.	Método Monte Carlo . . . . .	26
2.6.	Cadenas de Markov . . . . .	26
2.7.	Markov Chain Monte Carlo . . . . .	27
2.8.	Métricas de evaluación . . . . .	27
2.8.1.	Métricas puntuales . . . . .	27
2.8.1.1.	Error medio absoluto . . . . .	28
2.8.1.2.	Error medio absoluto escalado . . . . .	28
2.8.1.3.	Error porcentual medio absoluto . . . . .	28
2.8.1.4.	Error porcentual medio absoluto simétrico . . . . .	28
2.8.2.	Métricas probabilísticas . . . . .	29
2.8.2.1.	Probabilidad de cobertura del intervalo de predicción . . . . .	29
2.8.2.2.	Puntuación media del intervalo . . . . .	29

2.8.2.3.	Función de pérdida cuantílica escalada . . . . .	30
2.8.2.4.	Puntuación continua de probabilidad escalada . . . . .	30
<b>3.</b>	<b>Metodología</b>	<b>32</b>
3.1.	Datos y preprocesamiento . . . . .	32
3.1.1.	Características por tipo de supernova . . . . .	33
3.2.	Aumento de datos . . . . .	36
3.2.1.	Generación de curvas sintéticas . . . . .	36
3.2.2.	Perturbación de curvas reales . . . . .	38
3.3.	Balanceo de clases . . . . .	40
3.3.1.	Conjunto de entrenamiento . . . . .	40
3.3.2.	Conjunto de validación . . . . .	40
3.3.3.	Conjunto de prueba . . . . .	41
3.4.	Imputación de datos . . . . .	41
3.5.	Modelos Base . . . . .	42
3.5.1.	ALeRCE . . . . .	43
3.5.2.	Naïve . . . . .	43
3.6.	Modelo . . . . .	43
3.6.1.	Entrenamiento . . . . .	44
3.6.1.1.	Entradas . . . . .	44
3.6.1.2.	Valor deseado . . . . .	45
3.6.1.3.	Función de pérdida . . . . .	46
3.6.1.4.	Optimización . . . . .	47
3.6.2.	Predicción . . . . .	47
3.6.2.1.	Predicciones probabilísticas . . . . .	47
3.6.2.2.	Propuesta para considerar error de observación . . . . .	49
3.7.	Evaluación . . . . .	49
3.7.1.	Estrategia número de puntos . . . . .	50
3.7.2.	Estrategia umbral de días . . . . .	51
3.7.3.	Métricas . . . . .	51
3.7.4.	Evaluación para 5 conjuntos de datos diferentes . . . . .	51
<b>4.</b>	<b>Resultados, análisis y discusión</b>	<b>53</b>
4.1.	Comparación de modelos . . . . .	53
4.1.1.	Especificaciones del entrenamiento . . . . .	54

4.1.2.	Supernovas de tipo Ia . . . . .	54
4.1.2.1.	MASE . . . . .	55
4.1.2.2.	sMAPE . . . . .	55
4.1.2.3.	PICP y MSIS . . . . .	55
4.1.2.4.	CRPSS y SPL . . . . .	56
4.1.3.	Resumen de las métricas para SNIa . . . . .	57
4.1.4.	Resumen de las métricas para las clases SNII, SNIbc y SLSN . . . . .	57
4.2.	Predicciones . . . . .	58
4.2.1.	Predicciones con una observación . . . . .	60
4.2.2.	Ejemplo de predicción para una SNIa . . . . .	60
<b>5.</b>	<b>Conclusión</b>	<b>63</b>
5.1.	Trabajo Futuro . . . . .	64
5.1.1.	Modelos base para comparar . . . . .	64
5.1.2.	Horizonte y pasos de predicción . . . . .	64
5.1.3.	Búsqueda de hiperparámetros . . . . .	65
5.1.4.	Utilización de más covariables . . . . .	65
5.1.5.	Irregularidad en el muestreo de los datos . . . . .	65
5.1.6.	Complementar tareas de clasificación o detección de anomalías . . . . .	66
5.1.6.1.	Mejoras en tareas de clasificación . . . . .	66
5.1.6.2.	Tareas de detección de anomalías . . . . .	66
5.1.7.	Robustecer modelo de predicción . . . . .	66
5.1.8.	Utilización de arquitectura <i>Transformers</i> . . . . .	67
5.1.9.	Estimación de nuevos parámetros . . . . .	67
	<b>Bibliografía</b>	<b>76</b>
	<b>Anexos</b>	<b>77</b>
	<b>A. Métricas para SNII, SNIbc, SLSN</b>	<b>78</b>
	<b>B. Métricas según celda recurrente</b>	<b>85</b>
	<b>C. Ejemplos de predicciones</b>	<b>94</b>



# Índice de Tablas

3.1. Número de curvas válidas por tipo de supernova para cada una de las bandas	34
---	----

# Índice de Ilustraciones

1.1.	Imagen donde se observa la explosión de una supernova. La imagen de la izquierda muestra el instante anterior a la explosión de la supernova, mientras que la imagen de la derecha muestra la imagen con la supernova. Esta explosión se observa como un punto luminoso ubicado cerca del centro (ver círculo rojo). Imagen obtenida de <a href="https://news.fnal.gov/2013/04/supernovae-light-the-way-to-dark-energy/">https://news.fnal.gov/2013/04/supernovae-light-the-way-to-dark-energy/</a> . . . . .	2
2.1.	Representación de una estrella por capas según su elemento químico. Todos estos elementos son creados por la misma estrella a lo largo de su vida. Imagen sacada de <a href="https://ned.ipac.caltech.edu/level5/Sept16/Rauscher/Rauscher4.html">https://ned.ipac.caltech.edu/level5/Sept16/Rauscher/Rauscher4.html</a>	10
2.2.	Ejemplo de una curva de luz multibanda (ZTF21aaejeyu) de tipo SNIa obtenida por el sondeo ZTF. Se presentan las etapas más distintivas de la curva de luz de una supernova, dentro de las cuales se encuentra la zona de incremento, decaimiento por difusión y el decaimiento radioactivo. Además, se observan otras características como lo es el máximo brillo estimado y el inicio de lal decaimiento radioactivo. . . . .	12
2.3.	Taxonomía utilizada por el <i>broker</i> ALerCE para clasificar curvas de luz. La imagen fue obtenida de [1,2] . . . . .	13
2.4.	Esquema de categorización de series de tiempo con un ejemplo de los datos que se utilizarán. En naranja se marca la clasificación correspondiente a la curva de luz del ejemplo. . . . .	14
2.5.	Representación gráfica de los bloques más típicos de las redes neuronales recurrentes: LSTM y GRU. En ambos diagramas se especifica el nombre de cada compuerta con una representación de su composición. Además, se agrega una simbología en la parte inferior para entender de mejor forma el gráfico. La imagen fue obtenida <a href="https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21">https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21</a> . . . . .	19
3.1.	Gráfico de barra que compara el número de supernovas por tipo (SNIa, SNIi, SNIbc y SLSN) presentes en la base de datos provista por ALerCE - ZTF en su versión 8.0.3. . . . .	33
3.2.	Histogramas por cada tipo de supernova que muestra la cantidad de muestras detectadas tardíamente y tempranamente. Además, se agrega información sobre la mediana de detección temprana. . . . .	34

3.3. Distribución de diferencia de tiempo entre observaciones de todas las supernovas, luego de aplicar el preprocesamiento de los datos. La mediana de 3.0 días corresponde a la cadencia de ZTF. . . . .	35
3.4. Histogramas por cada tipo de supernova que muestran la cantidad de observaciones por curva de luz. Además, se agrega información sobre la mediana de observaciones. . . . .	35
3.5. Distribución de errores de observación del conjunto de entrenamiento para una partición de los datos. En esta se observa tanto el error de observación como el ajuste de su distribución. Además se presentan los valores asociados a dicha distribución. Finalmente, a partir de esta distribución se asignarán los errores de observación a cada uno de las observaciones sintéticas. . . . .	37
3.6. Ejemplo de aumento de datos sintéticos de la curva de luz ZTF20aabbihor de la banda r. La figura de arriba corresponde a la curva de luz observada, mientras que las 16 de abajo corresponden a variaciones obtenidas en base a la estrategia propuesta. . . . .	37
3.7. Esquema de aumento de datos sintéticos. A la izquierda se observan las estrategias de la etapa 1 y a la derecha las de la etapa 2. Cada línea representa el camino de variación que puede seguir cada curva de luz. . . . .	39
3.8. Ejemplo de aumento de datos sintéticos de la curva de luz ZTF20aabbihor de la banda r. La figura de arriba corresponde a la observación real, mientras que las 16 de abajo corresponden a variaciones obtenidas en base a la estrategia propuesta. . . . .	39
3.9. Proceso de imputación lineal de curvas de luz. (a) Muestra los datos observados, (b) muestra la discretización temporal centrada en 0 desde la primera observación en intervalos de 1 día, (c) muestra la imputación donde cada <i>bin</i> recibe la información de un punto. . . . .	42
3.10. Esquema de entrenamiento y predicción, donde a la izquierda (en azul) se muestra la arquitectura del codificador y a la derecha (en verde) se muestra la arquitectura del decodificador. En ambos casos, se estima una distribución normal con la cual se muestrea para generar una predicción. Al repetir el proceso de predicción usando el muestreo de Monte Carlo, se obtiene la estimación de la distribución de las predicciones. . . . .	45
3.11. Arquitectura del codificador junto al esquema de entrenamiento. La arquitectura utilizada consta de una RNN que obtiene los valores $\mu$ y $\sigma$ de una distribución normal. Para el caso de $\mu$ se obtiene a partir de una capa lineal, mientras que para $\sigma$ se le agrega una capa <i>softplus</i> para asegurar que este valor siempre sea positivo. Estos valores entran a una función de pérdida denominada <i>negative log likelihood</i> que se compara con el valor deseado $z(t)$ . . . . .	46

3.12. Arquitectura del decodificador junto al esquema de predicción. La arquitectura utilizada consta de una RNN que obtiene los valores $\mu$ y $\sigma$ de una distribución normal. Para el caso de $\mu$ se obtiene a partir de una capa lineal, mientras que para $\sigma$ se le agrega una capa <i>softplus</i> para asegurar que este valor siempre sea positivo. Estos valores permiten muestrear las predicciones de forma iterativa, a partir de una distribución normal $N(\mu, \sigma)$ . Estas predicciones ( $\hat{z}(t)$ ) variarán por cada ejecución de la predicción realizada. . . . .	48
3.13. Esquema de evaluación final. Los gráficos de la izquierda muestran los valores predichos en un horizonte de predicción de 7 días, a partir del último punto conocido. Los gráficos centrales muestran el proceso de ajuste de valores deseados con los que será comparada la predicción. Los gráficos de la derecha muestran la comparación realizada entre la predicción y los valores ajustados.	50
3.14. Ejemplos de estrategia umbral: En la figura se observan 3 curvas de luz que pasan por los 3 pasos de la estrategia de umbral de días. En el primer paso se aplica un umbral, el cuál se representa por la línea vertical azul. Posterior a eso, se determina el último punto (señalado con rojo) para cada una de las curvas de luz, donde se observa que no coinciden entre sí según el eje x. Finalmente, a partir del último punto se genera la predicción (puntos verdes) a un paso de 1 día con un horizonte de predicción de 7 días. . . . .	52
4.1. Conjunto de métricas en función del número de observaciones evaluadas solo en el conjunto de supernovas de tipo Ia. Cada métrica es obtenida mediante el promedio de los 5 k-folds cuya desviación estándar se representa como un área sombreada. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF all</b> (modelo entrenado con distintos tipos de supernovas) y <b>SNeF SNIa</b> (modelo entrenado solo con supernovas de tipo SNIa) . . . . .	58
4.2. Conjunto de métricas en función del umbral de días evaluadas solo en el conjunto de supernovas de tipo Ia. Cada métrica es obtenida mediante el promedio de los 5 k-folds cuya desviación estándar se representa como un área sombreada. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF all</b> (modelo entrenado con distintos tipos de supernovas) y <b>SNeF SNIa</b> (modelo entrenado solo con supernovas de tipo SNIa) . . . . .	59
4.3. Ejemplos de predicciones de 6 curvas de luz <b>distintas</b> cuando se conoce un solo punto. En esta figura se muestra la comparación entre la curva ajustada (utilizada para evaluar), la predicción propuesta por ALerCE y la predicción del modelo SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (Inicio de Predicción), donde el resto de observaciones solo son referenciales para comparar las predicciones . . . . .	61
4.4. Ejemplo de predicciones de una misma curva de luz pero con distinta cantidad de puntos como entrada. Se comparan las predicciones después de 1, 2, 4 y 6 puntos, respectivamente. También se muestra la predicción realizada por el modelo ALerCE-P. . . . .	62

A.1.	Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF-SNII</b> (modelo entrenado solo con supernovas de tipo SNII) y <b>SNeF - Covariables</b> (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	79
A.2.	Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF-SNII</b> (modelo entrenado solo con supernovas de tipo SNII) y <b>SNeF - Covariables</b> (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	80
A.3.	Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF-SNIbc</b> (modelo entrenado solo con supernovas de tipo SNIbc) y <b>SNeF - Covariables</b> (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	81
A.4.	Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF-SNIbc</b> (modelo entrenado solo con supernovas de tipo SNIbc) y <b>SNeF - Covariables</b> (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	82
A.5.	Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF-SLSN</b> (modelo entrenado solo con supernovas de tipo SLSN) y <b>SNeF - Covariables</b> (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	83
A.6.	Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF-SLSN</b> (modelo entrenado solo con supernovas de tipo SLSN) y <b>SNeF - Covariables</b> (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde a la mitad de la desviación estándar para simplificar la figura. . . . .	84

B.1. Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIa. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	86
B.2. Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNIa. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	87
B.3. Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	88
B.4. Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	89
B.5. Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	90
B.6. Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	91
B.7. Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALeRCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	92

B.8. Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALerCE-P, <b>SNeF - gru</b> y <b>SNeF - lstm</b> . Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura. . . . .	93
C.1. Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SNIa. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales. . . . .	94
C.2. Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SNII. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales. . . . .	95
C.3. Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SNIbc. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales. . . . .	96
C.4. Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SLSN. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales. . . . .	97

# 1. Introducción

## 1.1. Motivación

La astronomía es una ciencia centrada en la observación, que estudia el universo. Para estudiarlo, los astrónomos observan los distintos objetos y eventos astronómicos a partir de satélites, radiotelescopios y telescopios. Los telescopios poseen potentes cámaras que capturan imágenes del espacio. A partir del procesamiento de estas imágenes es posible obtener la variación del brillo en función del tiempo, lo cuál se manifiesta en las denominadas curvas de luz.

Con el paso del tiempo, la cantidad de datos que recopilan los telescopios ha ido en aumento. Este aumento ha obligado a los astrónomos a utilizar algoritmos de inteligencia computacional para detectar, clasificar, analizar y predecir los distintos astros de forma eficiente. Un claro ejemplo de un telescopio que generará una gran cantidad de datos corresponde al Vera C. Rubin a través del sondeo LSST (*Legacy Survey Space and Time*). Este sondeo, mapeará el cielo del hemisferio sur en tan solo 3 días, repitiendo esta acción por 10 años. Esto generará masivas cantidades de información que no podrán ser estudiadas directamente por humanos.

Las técnicas de aprendizaje automático corresponden a una excelente opción para procesar los datos provenientes de los telescopios. A partir de estas técnicas, es posible procesar las observaciones entrantes con el objetivo de adquirir la mayor información posible o cumplir tareas específicas.

En este trabajo de tesis se aborda la tarea de predecir el comportamiento futuro de las curvas de luz de un evento astronómico en particular: las supernovas (SNe). Una supernova es un evento astronómico transiente caracterizado por alcanzar brillos muy altos (Ver Figura 1.1) en periodos cortos de tiempo. Este brillo podría opacar incluso a las galaxias. Dada la característica anterior, los astrónomos han estudiado de cerca el comportamiento de las SNe para entender más sobre el universo. Algunos estudios buscan determinar las constantes cosmológicas, detectar galaxias que se encuentren muy lejanas, entender el comportamiento inicial del universo, entre otras.

Las supernovas tienen una corta duración en comparación a otros eventos astronómicos. Por lo anterior, detectarlas lo antes posible permitiría realizar un mejor seguimiento de éstas. Además, para obtener las mejores observaciones, los astrónomos deben realizar ajustes técnicos a los telescopios. Es por esto, que en esta tesis se propone una metodología para predecir el comportamiento futuro de una curva de luz junto a su incerteza. De esta forma, los astrónomos podrían realizar los ajustes asociados a estas predicciones e incertezas [3].





Figura 1.1: Imagen donde se observa la explosión de una supernova. La imagen de la izquierda muestra el instante anterior a la explosión de la supernova, mientras que la imagen de la derecha muestra la imagen con la supernova. Esta explosión se observa como un punto luminoso ubicado cerca del centro (ver círculo rojo). Imagen obtenida de <https://news.fnal.gov/2013/04/supernovae-light-the-way-to-dark-energy/>

Otras de las ventajas asociadas a la predicción, es que ésta podría ser utilizada como complementos para tareas de detección de artefactos y/o de clasificación.

## 1.2. Trabajos astronómicos relacionados

Las técnicas de aprendizaje computacional han sido utilizadas en astronomía para clasificar, detectar y analizar los distintos objetos y eventos astronómicos. Una tarea que ha sido poco explorada, en comparación al resto, corresponde a la de predecir el comportamiento de las curvas de luz. En el mejor de los esfuerzos, no se encontró literatura relacionada a la predicción de curvas de luz de supernovas.

En esta tesis, los trabajos relacionados a la predicción de series de tiempo utilizando RNNs son mostrados en la Sección 2.4.2 y los relacionados a la predicción de curvas de luz de astros se muestran en esta sección.

En la literatura se han desarrollado distintos métodos para enfrentar la tarea de predicción de curvas de luz de distintos astros. Existen trabajos que se centran en predecir un cierto evento de un astro en particular. Un ejemplo de lo anterior se encuentra en [4] donde se propone, mediante un análisis espectral, predecir el momento en el que ocurre un aumento repentino del brillo de esta estrella (*outbursts*). Este trabajo también propone una técnica para imputar los periodos de tiempo en los que no existen observaciones.

En [5] estudian la efectividad de los modelos autoregresivos en las series de tiempo astronómicas. En este se destacan los modelos autoregresivos de medias móviles (ARMA, por sus siglas en inglés), autoregresivos integrados de medias móviles (ARIMA, por sus siglas en inglés) y autoregresivos fraccionalmente integrados de medias móviles (ARFIMA, por sus

siglas en inglés). Además de los modelos anteriores se consideran versiones que manejan datos continuos en el tiempo e irregularmente muestreados. Este último modelo en particular es especialmente útil para trabajar sobre curvas de luz pues estas tienen asociado el problema de la irregularidad de los datos.

Por otro lado, existen trabajos que crean modelos que predicen el comportamiento futuro de una curva de luz (no necesariamente un evento) en base a modelos que utilizan redes neuronales. Estos modelos son entrenados en base a curvas de luz del mismo tipo de astro del que se desea predecir. En particular se suele utilizar redes neuronales recurrentes para la tarea de predecir por su capacidad de considerar datos secuenciales. En [6] se prueba el desempeño de las redes neuronales recurrentes LSTM con curvas de luz simuladas de núcleos activos galácticos. En este trabajo se estudia la sensibilidad de la función de pérdida de los conjuntos de entrenamiento y prueba ante el cambio de los distintos hiperparámetros de la red.

En [7, 8] se propone la utilización de redes neuronales recurrentes LSTM para predecir curvas de luz de la base de datos provista por el sondeo *Ground-based Wide-angle Camera array* (GWAC). En estos trabajos también se proponen distintos mecanismos para detectar anomalías en base al error de predicción. Por otro lado, en [9] se propone un método no paramétrico para modelar y predecir la variabilidad óptica de los *quasar* utilizando RNNs autocodificadoras. Este método es entrenado en base a curvas de luz provenientes del sondeo *Catalina Real-Time Transient Survey*.

Además de las redes neuronales recurrentes existen otras arquitecturas para generar predicciones. En [10] se propone un enfoque para predecir los datos faltantes y los futuros puntos de curvas de luz utilizando procesos gaussianos. Por otro lado, en [11] se propone una metodología para detectar anomalías en sondeos de transientes de larga escala utilizando redes neuronales probabilísticas construidas en base a redes convolucionales temporales. Para detectar estas anomalías se compara la predicción de un modelo con el valor observado. Dado lo anterior, implícitamente este modelo permitiría predecir curvas de luz. En [12] se presenta un detector de anomalías compuesto de 3 algoritmos de detección de anomalías para encontrar fuentes transientes, periódicas y estocásticas anómalas. El marco experimental del trabajo anterior se basa en la validación cruzada de seis algoritmos de detección de anomalías para cada clase mencionada (transiente, periódica y estocástica). También en [13] se proponen técnicas de aprendizaje profundo para detectar anomalías en los núcleos activos de galaxias (AGNs). Estos dos últimos trabajos utilizan la base de datos obtenida por el sondeo ZTF (*Zwicky Transient Facility*).

Uno de los trabajos que se utiliza como línea de base para comparar los resultados del predictor se basa en el trabajo de [14]. El foco principal de este trabajo corresponde a la clasificación, pero también proponen un modelo paramétrico de supernovas (SPM). A partir del ajuste de este modelo paramétrico de supernovas es posible generar predicciones <sup>1</sup>. Lo anterior es explicado en la Sección 3.5.1.

---

<sup>1</sup>API de ALerCE para predecir curvas de luz <https://api.alerce.online/forecast/v1>

## 1.3. Definición del problema

Los datos utilizados en este trabajo presentan una serie de dificultades que afectarían negativamente al desempeño del modelo predictivo:

- Escasez y desbalance de muestras de los distintos tipos de supernova
- Irregularidad en el muestreo de los datos, junto a valores faltantes para el caso de curvas de luz multibanda.
- No existe estacionalidad ni periodicidad, las cuales suelen ser características claves para la tarea de predicción de series de tiempo.

Las dificultades señaladas anteriormente, no permiten utilizar directamente los modelos conocidos de predicción. Es por esto, que se debe elaborar una serie de estrategias para aumentar la cantidad de datos, disminuir la irregularidad en el muestreo, buscar formas de aprender patrones basado en el aprendizaje cruzado entre las distintas muestras, entre otras.

## 1.4. Hipótesis

Las hipótesis que se barajan para la tarea de predicción son las siguientes:

1. Es posible generar predicciones probabilísticas de las curvas de luz de distintos tipos de supernovas en base a las observaciones conocidas, covariables y el aprendizaje cruzado formado al entrenar el modelo con distintas curvas de supernovas mediante redes neuronales recurrentes.
2. A partir del error de observación es posible generar estrategias que mejoren el entrenamiento y predicción del modelo predictivo.
3. A partir del modelo paramétrico de SNe es posible generar estrategias para mejorar el entrenamiento y evaluación del modelo predictivo. Se espera que mediante el ajuste del modelo paramétrico se puedan realizar evaluaciones en tiempos en los que no se tengan observaciones conocidas.

## 1.5. Objetivos

A continuación se describe el objetivo general y los objetivos específicos desarrollados en este trabajo de tesis.

### 1.5.1. Objetivo general

Desarrollar e implementar un predictor probabilístico de curvas de luz de supernovas de distintas clases, provenientes del sondeo ZTF, utilizando redes neuronales recurrentes.

Además se busca desarrollar e implementar estrategias para utilizar el error de observación y un modelo paramétrico de supernovas con el fin de robustecer el entrenamiento, predicción y evaluación.

### 1.5.2. Objetivos específicos

- Desarrollar e implementar un modelo que utilice redes neuronales recurrentes para generar predicciones probabilísticas de las curvas de luz de 4 tipos de supernovas.
- Desarrollar estrategias que utilicen la información del error de observación para mejorar el entrenamiento y la predicción del modelo.
- Desarrollar estrategias que utilicen la información del modelo paramétrico para mejorar el entrenamiento y poder generar estrategias de evaluación del desempeño de los modelos.
- Determinar covariables asociadas a las supernovas que permitan entregar información adicional al predictor.
- Evaluar el desempeño del predictor para cada tipo de supernova en base a las distintas variaciones propuestas y comparar resultados con el modelo de ALerCE.

## 1.6. Principales contribuciones

- Se desarrolla e implementa uno de los primeros modelos de predicción de curvas de luz de supernovas. En particular, el modelo propuesto es el primer modelo de predicción probabilística que utiliza aprendizaje profundo para predecir el comportamiento futuro de las curvas de luz de supernovas.
- Se proponen e implementan estrategias para considerar las incertezas asociadas al error de observación. Las cuales son utilizadas para mejorar el entrenamiento y las predicciones del modelo.
- Se proponen e implementan estrategias de evaluación para curvas de luz de supernovas utilizando el modelo paramétrico de supernovas.

## 1.7. Estructura de la Tesis

A continuación se presenta el listado de los capítulos de la presente tesis junto a una breve descripción de lo que tratará cada uno.

- **Antecedentes** (Capítulo 2): En este capítulo se presentan los conceptos básicos que serán tratados en este documento.
- **Metodología** (Capítulo 3): En este capítulo se describen los pasos a seguir para desarrollar el predictor de supernovas.

- **Resultados, análisis y discusión** (Capítulo 4): En este capítulo se muestran los resultados de los experimentos realizados, junto a sus respectivos análisis y discusiones.
- **Conclusiones** (Capítulo 5): En este capítulo se resume el trabajo realizado junto con presentar el grado de cumplimiento de los objetivos propuestos. Este capítulo además tiene una sección de trabajo futuro en donde se detalla cuales son los trabajos futuros propuestos que surgen en base al desarrollo de esta tesis.
- **Bibliografía**: Lista de referencias utilizadas en esta tesis.
- **Anexos**: Se anexa toda la información que complementa a la presente tesis pero que no corresponde al foco principal de ésta.

## 1.8. Acrónimos

A continuación se muestra un listado de los acrónimos más utilizados en esta tesis:

- AGNs: *Active Galactic Nuclei*.
- ALERCE: *Automatic Learning for the Rapid Classification of Events*.
- ANN: *Artificial Neural Network*.
- CRPS: *Continuous Ranked Probability Score*.
- GRU: *Gated Recurrent Unit*.
- LSST: *Legacy Survey Space and Time*.
- LSTM: *Long Short Term Memory*.
- MASE: *Mean absolute square error*.
- MAPE: *Mean absolute percentage error*.
- MCMC: *Markov Chain Monte Carlo*.
- MLP: *Multi Layer Perceptron*.
- MSIS: *Mean Scaled Interval Score*.
- PICP: *Prediction Interval Coverage Probability*.
- RNN: *Recurrent Neural Network*.
- SNe: *Supernovae*.
- SNeF: *Supernovae Forecaster*.
- SNIa: Supernova de tipo Ia.
- SNIbc: Supernova de tipo Ibc.
- SNII: Supernova de tipo II.

- SLSN: Supernova superluminosa.
- SPM: *Supernovae Parametric Model*.
- SPL: *Scaled Pinball Loss*.
- ZTF: *Zwicky Transient Facility*.

## 2. Antecedentes

### 2.1. Astronomía

En esta sección se presentan los conceptos astronómicos utilizados en esta tesis. Primero se abordan los conceptos de magnitud aparente, flujo fotométrico y como transformar de magnitud aparente a flujo. Luego, se explica qué es una supernova, objeto clave de este trabajo, junto a la definición de curva de luz. A partir de lo anterior, se describe un modelo paramétrico de supernovas que es parte fundamental del trabajo. Finalmente, se explica lo que es un sondeo astronómico y se presenta el sondeo del cuál se extrajo la información para el presente trabajo.

#### 2.1.1. Magnitud aparente

La magnitud aparente es una medida de la luminosidad de un objeto celeste. Se describe matemáticamente como la medición logarítmica inversa al brillo de una estrella percibido desde la Tierra. Lo anterior es representado en la siguiente ecuación:

$$\begin{aligned} m &= -2,5 \log_{10}\left(\frac{F_{estrella}}{F_{referencia}}\right) \\ &= -2,5 \log_{10}(F_{estrella}) + z_p, \end{aligned} \tag{2.1}$$

donde  $F_{estrella}$  corresponde al flujo fotométrico de la estrella y  $F_{referencia}$  corresponde al flujo fotométrico de referencia. Este último factor puede ser reemplazado por  $z_p$  que corresponde al punto cero. Este término incluye la información de la  $F_{referencia}$  a través del estudio de la difracción y absorción de color producida en la atmósfera.

#### 2.1.2. Flujo fotométrico

El flujo fotométrico, al igual que la magnitud aparente, es una medida del brillo de un objeto celeste. Este es medido según la cantidad de fotones recibidos por el dispositivo de carga acoplada (CCD; por sus siglas en inglés) de la cámara.

Dado que las mediciones de estas fuentes pueden presentar perturbaciones externas: clima, brillo de luna, presencia de satélites, etc, es posible calcular el error de observación.

### 2.1.3. Transformación de magnitud a flujo

La información de la luminosidad de las curvas de luz de supernovas provistas por ALerCE-ZTF se encuentran en magnitud. Estas serán transformadas a flujo a partir de la siguiente ecuación:

$$F_{estrella}(m) = k \cdot 10^{-\frac{(mag+z_p)}{2,5}}, \quad (2.2)$$

$$\sigma_{F_{estrella}}(m, \sigma_m) = k \cdot \left(10^{-\frac{(mag+z_p)}{2,5}} - 10^{-\frac{(mag+\sigma_m)+z_p}{2,5}}\right), \quad (2.3)$$

donde  $F_{estrella}$  corresponde al flujo de la estrella,  $mag$  corresponde a la magnitud aparente,  $z_p$  corresponde al punto cero,  $\sigma_m$  corresponde al error de observación en magnitud,  $\sigma_{F_{estrella}}$  corresponde al error de observación en flujo, y  $k$  es una constante utilizada para escalar los datos

### 2.1.4. Supernovas

Las supernovas (SNe) son eventos astronómicos transientes que nacen en base a una explosión de una estrella. Estas se caracterizan por alcanzar grandes brillos en cortos periodos de tiempo. Según como se generan estas explosiones y/o según su espectro electromagnético se pueden clasificar en varios tipos de supernovas: SNIa, SNII, SNIbc, SNIIn, SLSN, entre otras.

Las supernovas que nacen por el colapso del núcleo de la estrella corresponden principalmente a SNIbc, SLSN y SNII. Para las SNII, este colapso ocurre por los cambios en la estructura y composición de la estrella durante su ciclo de vida. Primero, el hidrógeno que posee la estrella es transmutado en helio liberando así grandes cantidades de energía en forma de radiación. Luego, en estrellas masivas, el helio transmuta a elementos más pesados como el carbono, oxígeno y hierro. A medida que la estrella transmuta hacia elementos más pesados (Ver Figura 2.1), estos disminuyen la presión interna de la estrella lo que provoca un desequilibrio de las fuerzas que mantienen estable a una estrella. Este desequilibrio provoca que la estrella colapse bajo su propio peso. Debido a lo anterior, las capas más externas de las estrellas son expulsadas rápidamente en una gran explosión.

Por otro lado, existen supernovas, como las de tipo Ia, que nacen por la explosión de enanas blancas que se encuentran en sistemas binarios. Las enanas blancas incrementan su masa al absorber el material de la otra estrella dentro del sistema. Cuando la enana blanca alcanza el límite de *Chandrasekhar* ocurre una reacción termonuclear que provoca que explote.

Dada la gran luminosidad que alcanzan las supernovas se utilizan como faroles del espacio que sirven como indicadores de distancia. A partir de la detección de una de ellas, se puede determinar la existencia galaxias que en principio, por su baja luminosidad, no se observan. Además las supernovas son utilizadas para calcular constantes cosmológicas y para estudiar las condiciones iniciales del universo.

### 2.1.5. Modelo paramétrico de supernovas

El modelo paramétrico de supernovas (SPM; por sus siglas en inglés) es una función analítica que describe el comportamiento de una supernova en función de sus datos. En



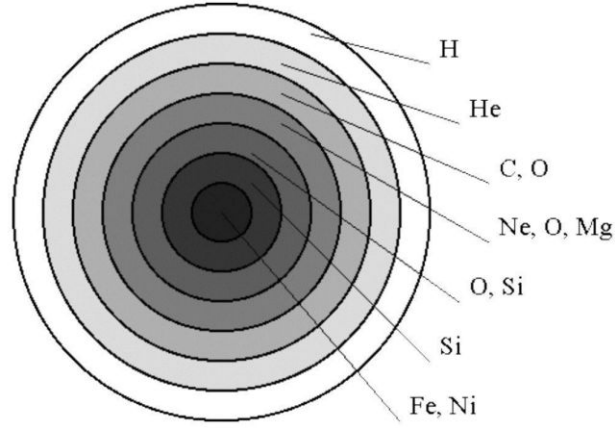


Figura 2.1: Representación de una estrella por capas según su elemento químico. Todos estos elementos son creados por la misma estrella a lo largo de su vida. Imagen sacada de <https://ned.ipac.caltech.edu/level5/Sept16/Rauscher/Rauscher4.html>

la literatura se han propuestos diferentes modelos paramétricos con distintos parámetros ajustables [14, 15, 16, 17, 18].

En este trabajo, se utilizará el SPM propuesto por [14], el que describe en la siguiente ecuación:

$$F = A \underbrace{\frac{1 - \beta' \frac{t-t_0}{t_1-t_0}}{1 + \exp\left(-\frac{t-t_0}{\tau_{rise}}\right)} \left[1 - \sigma\left(\frac{t-t_1}{5}\right)\right]}_{\text{temprano}} + A \underbrace{\frac{(1 - \beta') \cdot \exp\left(\frac{t-t_1}{t_{fall}}\right)}{1 + \exp\left(-\frac{t-t_0}{\tau_{rise}}\right)} \left[\sigma\left(\frac{t-t_1}{5}\right)\right]}_{\text{tardío}}. \quad (2.4)$$

Tal como se observa en la Ec. (2.4), la función se divide en dos. La primera parte (temprano) domina el flujo temprano de la curva de luz, mientras que la segunda parte (tardío) domina el flujo tardío. Además, si se compara esta función con la propuesta por [14] se encuentra una leve diferencia en el factor de suavizado (denominador dentro de la función sigmoide  $\sigma$ ) de la transición. Esto se debe a que se utiliza la propuesta mostrada en [19].

Los parámetros junto al efecto que producen dentro del modelo se presentan en el siguiente listado:

- $A$ : Afecta la escala del brillo de la supernova
- $\beta'$ : Controla la pendiente de la meseta
- $t_0$ : Actúa como un desplazador temporal de la curva de luz. Este suele acercarse al momento del brillo máximo
- $t_{rise}$ : Controla el tiempo para alcanzar el máximo brillo de la curva de luz
- $t_1$ : Controla la duración de la meseta
- $t_{fall}$ : Controla la duración del decaimiento
- $\sigma$ : Función sigmoide.

### 2.1.6. Curvas de luz

Las curvas de luz son series de tiempo astronómicas que muestran el brillo de una fuente luminosa en función de su tiempo [20]. Estas se obtienen a través de *surveys* astronómicos (ver Sección 2.1.7). Cada medición realizada proporciona información del brillo, error, tiempo de detección y las bandas fotométricas asociadas a la observación.

En este trabajo se estudia el comportamiento de las curvas de luz de supernovas, las cuales se caracterizan por poseer un rápido incremento en su brillo y un posterior decaimiento de este. En la Figura 2.2 se observa el comportamiento típico de una supernova (de tipo Ia), junto a algunas de las características asociadas a estas. A continuación se detalla cada una de estas características:

- Largo variable: Dada la naturaleza de las mediciones y del evento astronómico, las curvas de luz suelen tener un largo variable.
- Error de observación: Dada la naturaleza de las mediciones, estas observaciones pueden estar contaminadas por factores externos. Algunos de estos factores corresponden a factores atmosféricos como la existencia de gotas de agua, nubes, etc. También las observaciones pueden verse afectadas por el brillo de otras fuentes como la Luna, el paso de satélites, entre otros.
- Cadencia irregular: Dada la naturaleza de las mediciones, la observación del evento astronómico puede verse afectado por efectos climáticos, humanos y/o factores técnicos. Esto podría no solo inducir un error de observación, sino que podría evitar la medición de la fuente en estudio. Lo anterior puede provocar que existan grandes periodos de tiempo sin observaciones y dificultar el correcto seguimiento de la fuente.
- Primera observación: En principio, no es posible determinar con exactitud si la primera observación realizada corresponde a un instante anterior o posterior al máximo brillo alcanzado. Dado lo anterior, el análisis de las curvas de luz suele complejizarse.
- Datos faltantes entre bandas: Dada la naturaleza de la medición, no es posible obtener la información de todas las bandas en exactamente la misma estampilla temporal. A pesar de que el tiempo entre observaciones de distintas bandas podría ser bajo, no siempre se tendrá la información para todas las bandas.
- No son periódicas ni estacionales: En las tareas de predicción se suele aprovechar el comportamiento periódico y/o estacional de las series de tiempo en estudio. Este comportamiento permitiría predecir el futuro en base a la información pasada de la misma. Pero en el caso de las supernovas, al ser eventos transientes éstas no presentarían periodicidad ni estacionalidad [20].

### 2.1.7. Sondeos, conjuntos de datos y *brokers*

Los sondeos astronómicos son planes de acción que estudian determinados objetos o eventos de interés. Para realizar lo anterior se define el objeto u evento, la región a mapear, la cadencia, el tiempo de observación, los instrumentos a utilizar, las bandas que se medirán, entre otros.

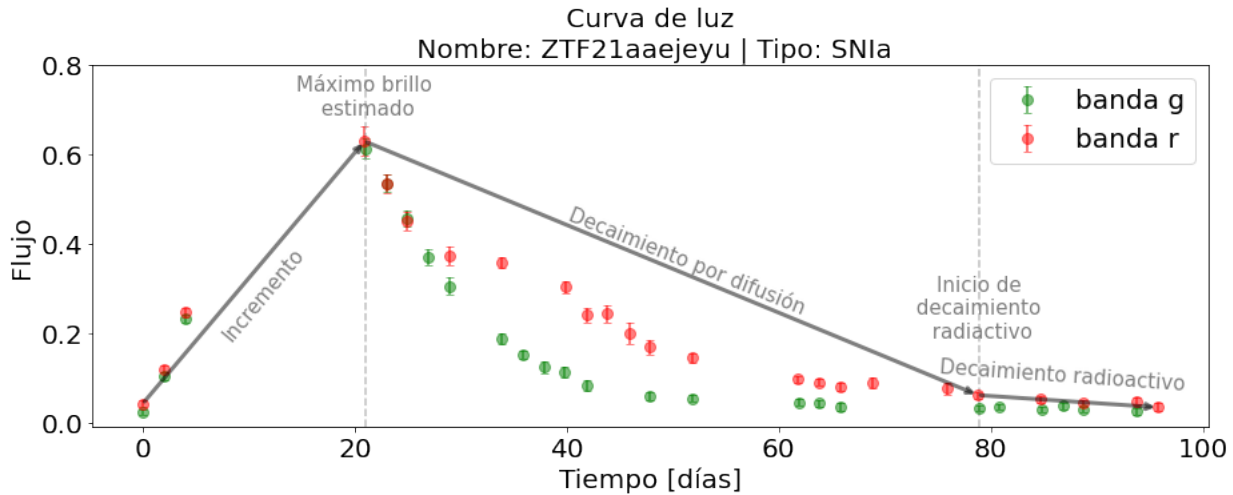


Figura 2.2: Ejemplo de una curva de luz multibanda (ZTF21aaejeyu) de tipo SNIa obtenida por el sondeo ZTF. Se presentan las etapas más distintivas de la curva de luz de una supernova, dentro de las cuales se encuentra la zona de incremento, decaimiento por difusión y el decaimiento radioactivo. Además, se observan otras características como lo es el máximo brillo estimado y el inicio de la decaimiento radioactivo.

A partir de estos sondeos se arman catálogos astronómicos que son fuentes útiles que mapean la información de los distintos astros del universo. Además, las observaciones realizadas por estos sondeos, se utilizan para desarrollar modelos de detección, clasificación, análisis y/o predicción.

A continuación se presentan 3 sondeos que están relacionados con el estudio de objetos transientes:

- *High Cadence Transient Survey* (HiTS; [21]): Sondeo diseñado con el objetivo de detectar el fenómeno conocido como *shock breakout*, el cuál corresponde a la onda de choque que se produce cuando la presión interna de la estrella es vencida por la gravedad. Otro de sus objetivos fue realizar un seguimiento de objetos astronómicos transientes como lo son las supernovas. Este sondeo tuvo una cadencia cercana a las 2 horas donde se observaron 40 campos del cielo en una banda fotométrica (banda  $u$ ).
- *Zwicky Transient Facility* (ZTF; [22]): Este sondeo, al igual que el anterior, tiene como objetivo estudiar eventos astronómicos transientes. Este se caracteriza por poseer una alta cadencia, capturando el cielo observable del hemisferio norte del planeta cada 3 noches. Las bandas utilizadas corresponden a  $g$  y  $r$ .
- *Legacy Survey of Space and Time* (LSST; [23]): Este sondeo, al igual que los anteriores tiene como objetivo estudiar eventos astronómicos transientes, donde se espera detectar un gran número de supernovas. Este se caracteriza por poseer una cadencia de 15 días y busca mapear todo el cielo nocturno. Para este caso, se realizan observaciones en 6 bandas distintas. Además, está diseñado para estudiar una gran porción (hemisferio sur observable) del cielo nocturno en 6 bandas distintas. Se estima que este sondeo observará millones de alertas transientes por noche [11].

Dado que los sondeos más actuales (y los próximos) son (serán) capaces de recolectar

masivas cantidades de información, se requiere obtener formas ordenadas para manejar los datos. Con el objetivo de recolectar, analizar e identificar astros de interés, nacen los denominados *brokers* astronómicos. Un *broker* es una fuente intermediaria entre el telescopio y la comunidad científica.

El *Automatic Learning for the Rapid Classification of Events* (ALeRCE; [1,2]) es un *broker* Chileno que procesa en tiempo real la información capturada por el sondeo ZTF. ALeRCE automáticamente extrae características astrofísicas que son aprovechadas para clasificar en tres grandes categorías: objetos transientes, variables y estocásticos ([24]). En la Figura 2.3 se observa un esquema con la taxonomía usada por ALeRCE que clasificar entre los distintos tipos de curvas de luz.

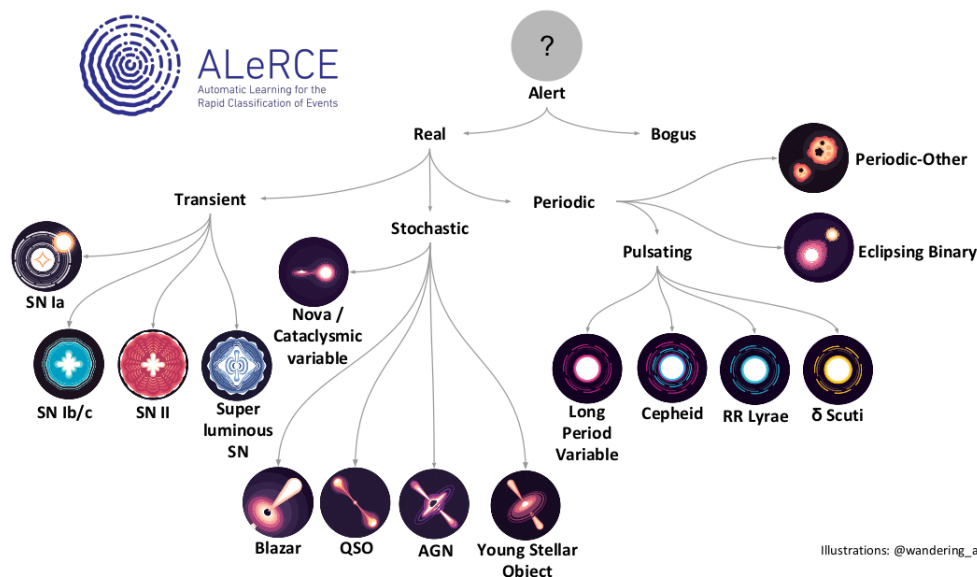


Figura 2.3: Taxonomía utilizada por el *broker* ALeRCE para clasificar curvas de luz. La imagen fue obtenida de [1,2]

## 2.2. Series de tiempo

Las series de tiempo son colecciones de observaciones obtenidas secuencialmente en el tiempo. Estas pueden ser clasificadas según su continuidad, cantidad de variables de interés, muestreo, entre otros. Un resumen de los distintos tipos de categorías se presenta en la Figura 2.4 la que se detallan en las siguientes secciones.

### 2.2.1. Series continuas o discretas

Las series de tiempo pueden ser clasificadas según su continuidad como: continuas o discretas [25]. Las series temporales continuas son obtenidas cuando las observaciones son hechas continuamente sobre un intervalo de tiempo. Por otro lado, las series de tiempo discretas se obtienen cuando las observaciones son hechas en un conjunto discreto de datos

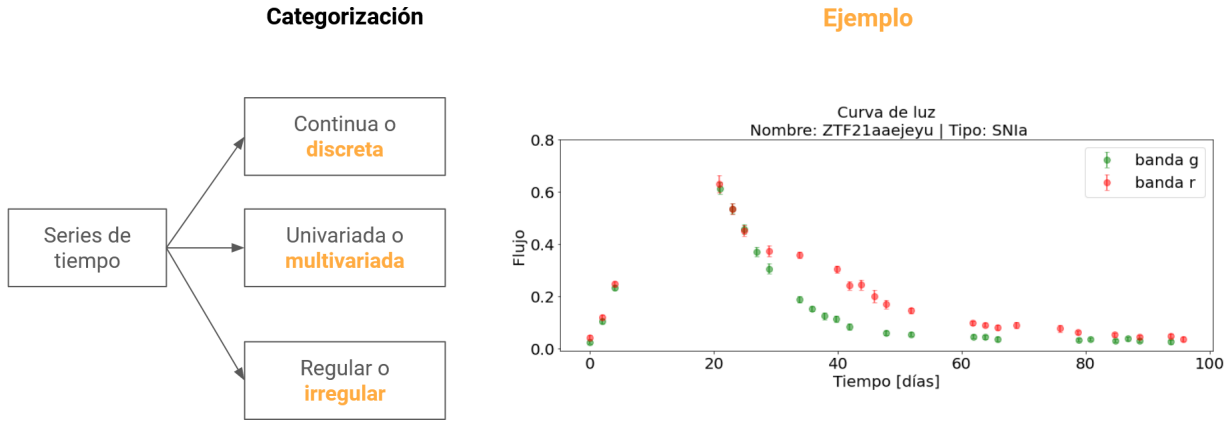


Figura 2.4: Esquema de categorización de series de tiempo con un ejemplo de los datos que se utilizarán. En naranja se marca la clasificación correspondiente a la curva de luz del ejemplo.

[26]. En este trabajo, se analizan observaciones provenientes de un telescopio, que por su naturaleza, son discretas.

### 2.2.2. Series univariadas y multivariadas

Las series de tiempo pueden ser clasificadas según la cantidad de variables que son medidas en cada instante de tiempo como: univariadas y multivariadas [27]. Las series univariadas poseen una sola observación de interés por estampilla de tiempo. Por otro lado, las series de tiempo multivariadas recopilan la información de más de una variable de interés en una misma estampilla de tiempo [28]. En este trabajo, se analizan las observaciones obtenidas por el telescopio en dos bandas. La información de cada banda se entrega de forma separada pero sobre el mismo objeto. Es por esto, que las series de tiempo tratadas en este trabajo corresponden a series multivariadas.

### 2.2.3. Series regularmente e irregularmente muestreadas

Las series de tiempo pueden ser clasificadas según la regularidad en el muestreo de los datos como series regularmente muestreadas e irregularmente muestreadas. Las series de tiempo regularmente muestreadas se obtienen cuando las observaciones se realizan en intervalos equiespaciados de tiempo. Por otro lado, las series de tiempo irregularmente muestreadas se obtienen cuando las observaciones son muestreadas en intervalos no equiespaciados en el tiempo. Este último tipo de series abundan en astronomía y medicina debido a la forma en la que se mide. Este tipo de mediciones dependen de distintas condiciones para llevarse a cabo. Por ejemplo en astronomía, para que una observación sea considerada, se requiere que el error de observación sea al menos 5 veces menor que su brillo. Es por lo anterior, que las series de tiempo de esta tesis corresponden series irregularmente muestreadas.

## 2.2.4. Series con valores faltantes

Un problema asociado a las series de tiempo regulares univariadas es que la obtención de la medición falle, provocando así una pérdida de datos. Según el tipo de análisis que se utilice, se debe determinar si este valor faltante será ignorado, imputado o abordado con alguna otra técnica.

Un problema similar ocurre para las series de tiempo multivariadas. Ya sea una serie regular o irregular, se puede dar el caso en que no se pueda obtener la información de todas las variables a la vez. Lo anterior correspondería a una serie de tiempo multivariada con valores faltantes. Al igual que en el caso anterior, según el análisis que se realice se debe tomar la decisión de ignorar, imputar o abordar con alguna otra técnica este valor faltante.

En este trabajo se revisan series de tiempo irregularmente muestreadas con valores faltantes. Este tipo de series de tiempo limitan severamente la habilidad de ser analizadas por técnicas aprendizaje automático [27].

## 2.3. Aprendizaje profundo

El presente trabajo utiliza algoritmos de aprendizaje profundo para llevar a cabo la tarea de predicción de series de tiempo. Dado lo anterior, en esta sección se detallan los conceptos asociados a este tipo de aprendizaje. Para esto, se explica primero que es una red neuronal artificial. Posterior a ello se detalla que son las redes neuronales recurrentes junto a sus celdas típicas. Finalmente, se explica la forma en la que estas redes aprenden y como evitar el sobreajuste de los datos.

### 2.3.1. Redes neuronales artificiales

Las redes neuronales artificiales (ANN, por sus siglas en inglés) son modelos matemáticos inspirados en las neuronas biológicas. Las ANN poseen unidades básicas denominadas neuronas. Estas presentan conexiones de entradas que serán procesadas para generar una salida. La forma más simple de esta red neuronal artificial se denomina neurona o perceptrón y se define matemáticamente como:

$$y = \phi(W^T x + b), \quad (2.5)$$

donde  $W = [w_1, \dots, w_n]$  es el vector de pesos o ponderadores,  $x = [x_1, \dots, x_n]$  es el vector de entradas compuesto por  $n$  elementos,  $b \in \mathbb{R}$  es el sesgo,  $y$  corresponde a la salida del modelo y  $f : \mathbb{R} \rightarrow \mathbb{R}$  se denomina función de activación, la cuál usualmente es no lineal.

A partir de la composición no lineal de perceptrones, es posible obtener el denominado perceptrón multicapa (MLP, por sus siglas en inglés; [29]). Este modelo es un aproximador universal capaz de estimar cualquier función no lineal si se tiene la suficiente información. Su representación matemática es la siguiente:

$$y_i = \phi_i(W_i^T x + b_i), \quad (2.6)$$

$$h(x) = (y_N \circ \dots \circ y_0)(x), \quad (2.7)$$

donde  $y_i$  es la salida de un perceptrón,  $\phi_i$  es la función de activación,  $W_i$  corresponden al vector de pesos,  $x$  corresponde a las entradas,  $b$  es el sesgo,  $h$  la salida de la MLP y  $\circ$  es el operador de composición de funciones.

### 2.3.2. Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN, por sus siglas en inglés; [29]) son un tipo de red neuronal artificial especializada en el manejo de datos secuenciales. Este tipo de datos se encuentran en textos, vídeos, audio y series de temporales. Esta red ha demostrado ser un aproximador universal [30] capaz de aproximar sistemas dinámicos si se tiene la suficiente información.

A continuación se describen 3 tipos de redes neuronales recurrentes. La primera corresponde a la *Vanilla RNN* que es una de las versiones más simples de esta red. Luego, se muestran dos versiones que se basan en la utilización de compuertas lógicas para mejorar el desempeño de la primera.

#### 2.3.2.1. Vanilla RNNs

Las versiones más simples de las RNNs se denominan *Vanilla RNN*. Estas se caracterizan por poseer un estado de memoria que colapsa la información en un vector denominado vector de contexto. Este vector es actualizado recursivamente según la cantidad de datos secuenciales presentes. Lo anterior permite trabajar con datos de largo variable. A continuación se presenta la ecuación del estado interno de la memoria:

$$z_t = \nu(z_{t-1}, y_t, x_t, s), \quad (2.8)$$

donde  $z$  corresponde al estado oculto interno de la RNN,  $\nu$  a la función de actualización de memoria,  $y_t$  al resultado de la capa anterior en el instante  $t$ ,  $x_t$  a la entrada en el tiempo,  $s$  a los parámetros de la función de actualización de memoria.

Un ejemplo de una *Vanilla RNN* corresponde a la Elman RNN [31], la cuál posee la siguiente forma:

$$y_{t+1} = \gamma_y(W_y z_t + b_t), \quad (2.9)$$

$$z_t = \gamma_z(W_{z_1} z_{t-1} + W_{z_2} y_t + W_{z_3} x_t + W_{z_4} s + b_z), \quad (2.10)$$

donde  $y_{t+1}$  es la salida de la red en el tiempo  $t+1$ ,  $z$  es el estado oculto interno en el instante  $t$ ,  $W$  es la matriz de pesos,  $b$  es el sesgo,  $\gamma$  es la función de activación.

Estas versiones de RNN poseen problemas de memoria, es decir, a medida que se avanza en la secuencia, esta tiende a olvidar la información del pasado. Lo anterior se produce por el desvanecimiento del gradiente, el cual es un problema que dificulta el entrenamiento de este tipo de redes. Este problema ocurre cuando los gradientes que se propagan hacia atrás se hacen muy pequeños, provocando que las actualizaciones de los pesos de las neuronas se vuelvan muy pequeñas.

Dado lo anterior, se propusieron diversas formas para abordar el desvanecimiento del gradiente. La forma más conocida consiste en utilizar compuertas lógicas que permitan retener, olvidar o actualizar la información. A continuación se presentan dos versiones de RNN basadas en compuertas lógicas.

### 2.3.2.2. Long Short-Term Memory

La *Long Short-Term Memory* (LSTM; [32]) es una red que soluciona los problemas de la versión básica de la red neuronal recurrente. Esta arquitectura asume que los datos pueden ser manejados por compuertas encargadas de olvidar y mantener información a lo largo del tiempo (ver esquema izquierdo de la Figura 2.5). Estas compuertas son gobernadas por las siguientes ecuaciones:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}), \quad (2.11)$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}), \quad (2.12)$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}), \quad (2.13)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}), \quad (2.14)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \quad (2.15)$$

$$h_t = o_t \odot \tanh(c_t), \quad (2.16)$$

donde  $x_t$  es la entrada en el instante  $t$ ,  $h_t$  corresponde al estado oculto,  $c_t$  corresponde a la celda,  $i_t$ ,  $f_t$ ,  $g_t$ ,  $o_t$  son las compuertas de entrada, olvido, celda propuesta, y salida.  $\sigma$  es la función sigmoide y  $\odot$  es el producto Hadamard elemento a elemento. Los términos  $W$  y  $b$  corresponden a los pesos y bias de las conexiones entre neuronas. A continuación se detalla el comportamiento de cada una de las compuertas

**Compuerta de entrada:** Más conocida como *input gate*, decide qué información será guardada en la memoria a largo plazo actualizando el estado de la celda. Para esto, la información del estado oculto anterior y la entrada actual pasan por una función sigmoide, obteniéndose  $i_t$ . Al igual que para la compuerta de olvido, el resultado de la función anterior esta entre 0 y 1. Lo anterior determina la importancia de la función tangente hiperbólica del estado oculto anterior y la entrada actual ( $g_t$ ), a través de la multiplicación punto a punto de estos ( $i_t \odot g_t$ ).

**Compuerta de olvido:** Más conocida como *forget gate*, decide que información debería ser olvidada. Para esto, la información del estado oculto anterior y la entrada actual pasan por una función sigmoide, obteniéndose  $f_t$ . Esta función tiene como salida valores entre 0 y 1. Mientras más cerca del 0, la información será olvidada y mientras más cerca del 1, la información se mantendrá.

**Compuerta de salida:** Más conocida como *output gate*, decide como debería ser el próximo estado oculto. Para esto, la información del estado oculto anterior y la entrada actual pasaran por una función sigmoide, obteniéndose  $o_t$ . Al igual que para las compuertas ya mencionadas, el resultado de la función anterior esta entre 0 y 1. Lo anterior determina la importancia de la función tangente hiperbólica de la celda ( $c_t$ ).

### 2.3.2.3. Gated Recurrent Unit (GRU)

La *Gated Recurrent Unit* (GRU; [33]) es otra versión de las RNN que, al igual que las redes LSTM, posee compuertas con funciones específicas (ver esquema derecho de la Figura 2.5). Estas compuertas son gobernadas por las siguientes ecuaciones:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}), \quad (2.17)$$



$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}), \quad (2.18)$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}), \quad (2.19)$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{t-1} + b_{hn})), \quad (2.20)$$

$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{t-1}, \quad (2.21)$$

donde  $x_t$  es la entrada en el instante  $t$ ,  $h_t$  corresponde al estado oculto,  $r_t$ ,  $z_t$ ,  $n_t$  son las compuertas de reseteo, actualización y la nueva.  $\sigma$  es la función sigmoide y  $\odot$  es el producto Hadamard. Los términos  $W$  y  $b$  corresponden a los pesos y bias de las conexiones entre neuronas. A continuación se detalla el comportamiento de cada una de las compuertas.

**Compuerta de reinicio:** Más conocida como *reset gate*, decide qué tanta información del pasado será olvidada. Para esto, la información del estado oculto anterior y la entrada actual pasan por una función sigmoide, obteniéndose  $r_t$ . Esta función es multiplicada punto a punto por el estado oculto anterior. Finalmente se calcula la tangente hiperbólica de la suma de la entrada actual y del resultado de la función anterior.

**Compuerta de actualización:** Más conocida como *update gate*, decide qué tanta información del pasado pasará a lo largo del estado. Para esto, la información del estado oculto anterior y la entrada actual pasan por una función sigmoide, obteniéndose  $z_t$ . Esta función es multiplicada punto a punto por el estado oculto anterior.

### 2.3.3. Entrenamiento de redes neuronales

La tarea de entrenamiento de una red neuronal consiste en resolver un problema de optimización para ajustar los parámetros de la red. Este problema usualmente es resuelto mediante la técnica denominada descenso del gradiente. Esta técnica busca minimizar el error de ajuste de los datos basados en una **función de costo**.

A continuación se describe de forma simplificada los pasos del algoritmo del descenso del gradiente:

- Se inicializan los parámetros de la red de forma aleatoria.
- Se calcula el gradiente de la función de costos en base a todos los parámetros de la red.
- Se actualizan los parámetros del modelo en dirección opuesta al gradiente, utilizando una **tasa de aprendizaje**. Esta tasa de aprendizaje es un hiper parámetro que define cuánto afectara el gradiente en el ajuste de los parámetros.
- Se repiten los últimos dos pasos hasta que los valores converjan.

Una limitante técnica del descenso del gradiente es que su costo computacional aumenta mientras más compleja sea la red. Para resolver lo anterior, se utiliza un algoritmo que retropropaga el error desde la capa de salida hasta la capa de entrada. Este algoritmo conocido como *backpropagation* [29] utiliza la regla de la cadena para calcular el gradiente en los parámetros de la red.

Para hacer más eficiente el cálculo del gradiente, se puede utilizar el algoritmo de gradiente descendente por lotes pequeños. Este consiste en dividir el conjunto de entrenamiento en mini

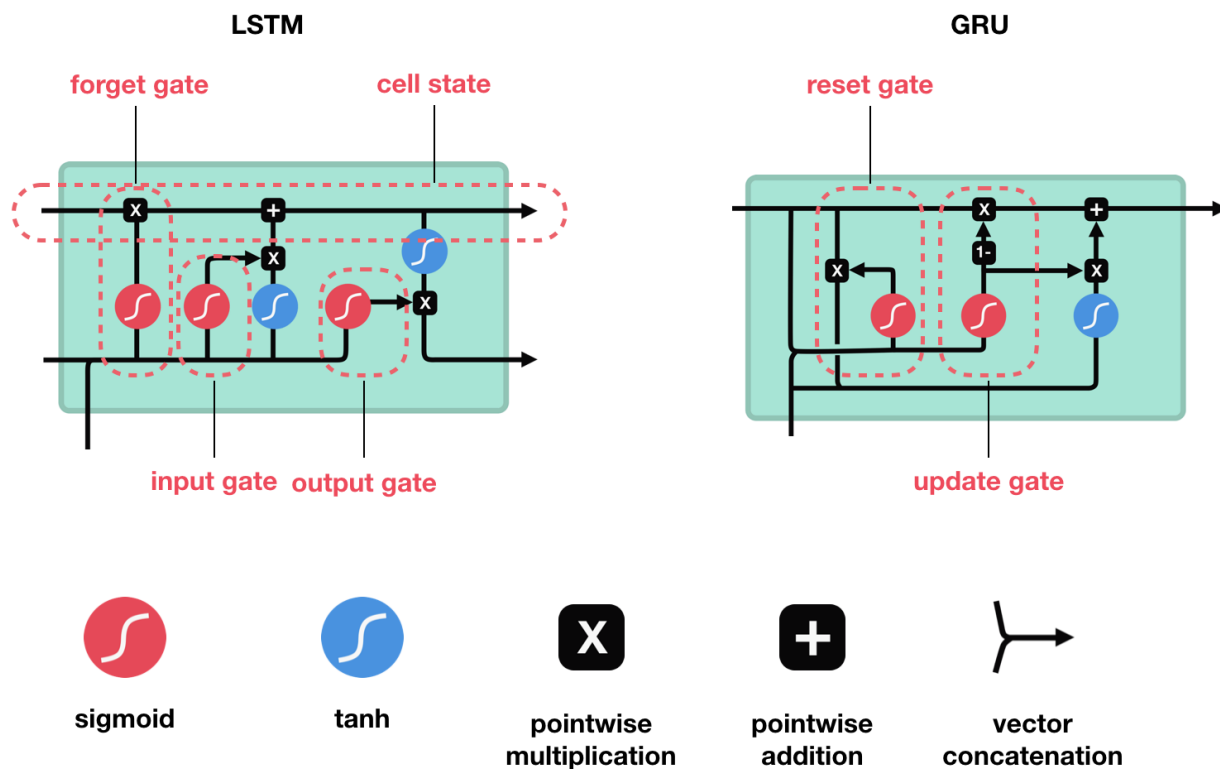


Figura 2.5: Representación gráfica de los bloques más típicos de las redes neuronales recurrentes: LSTM y GRU. En ambos diagramas se especifica el nombre de cada compuerta con una representación de su composición. Además, se agrega una simbología en la parte inferior para entender de mejor forma el gráfico. La imagen fue obtenida <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.

lotes de datos, con los cuales se entrenan y actualizan los parámetros. Esta técnica es una variación del gradiente descendente estocástico (SGD, por sus siglas en inglés).

En esta tesis, se utiliza una versión modificada del SGD denominada *Adaptive Moment Estimation* (Adam; [34]). Una de las grandes ventajas de utilizar esta técnica corresponde a su capacidad para ajustar la tasa de aprendizaje de forma adaptativa. Esta adaptabilidad se logra a través del cálculo de medias móviles exponenciales de los gradientes y de sus cuadrados. Esto permite controlar la actualización de los parámetros, evitando así alcanzar mínimos locales.

Los principales hiperparámetros del algoritmo Adam corresponden a la tasa de aprendizaje y a sus reguladores  $\beta_1$  y  $\beta_2$ . El coeficiente  $\beta_1$  es utilizado para calcular una estimación de la media de los gradientes pasados y el coeficiente  $\beta_2$  estima la varianza de estos gradientes.

### 2.3.3.1. Separación de base de datos

Con el objetivo de que el modelo pueda generalizar y cumplir la tarea requerida, es que se separa el conjunto de datos obtenidos en tres partes: entrenamiento, validación y prueba. A continuación se describen estos 3 conjuntos:

- Entrenamiento: Conjunto de datos que serán utilizados por el modelo para determinar los parámetros de la red.
- Validación: Conjunto de datos que serán utilizados para seleccionar el mejor modelo. Este conjunto además es utilizado para evitar problemas de sobreajuste del modelo.
- Prueba: Conjunto de datos que el modelo nunca antes ha visto. Este conjunto es utilizado para determinar qué tan bien generaliza el modelo.

## 2.3.4. Técnicas de regularización

Los modelos de aprendizaje profundo se caracterizan por presentar parámetros ajustables. Estos parámetros se suelen ajustar con el conjunto de datos entrenamiento. Si estos parámetros se ajustan a los datos de entrenamiento, el poder generalizador y/o de aprendizaje disminuye produciéndose un sobreajuste. Para evitar este problema, existe una diversidad de técnicas que permiten controlar o disminuir el riesgo de sobreajuste. A continuación se presentan una serie de regularizadores que fueron utilizados en la presente tesis.

### 2.3.4.1. Detención temprana

La detención temprana (más conocida como *Early Stopping*) es una técnica utilizada para evitar el sobreajuste en el entrenamiento. Esta técnica compara el comportamiento de la función de pérdida en los conjuntos de entrenamiento y validación. Cuando la función de pérdida del conjunto de validación deja de mejorar (mientras que la función del conjunto de entrenamiento sigue mejorando) el entrenamiento se detiene. De esta forma se evita que el modelo memorice el conjunto de entrenamiento.

Esta técnica se ajusta en base a la cantidad de épocas de entrenamiento. Si el modelo deja de mejorar durante una cantidad  $n$  de épocas, entonces el entrenamiento se detiene.

### 2.3.4.2. Perturbar datos de entrada

Agregar ruido a los datos de entradas es una técnica de regularización utilizada para evitar el sobreajuste en los modelos de aprendizaje [35]. Agregar ruido durante el proceso de entrenamiento permitiría al modelo aprender patrones robustos y generales de los datos, evitando así sobreajustarse a los datos.

### 2.3.4.3. Dropout

El *Dropout* [36] es una técnica utilizada para reducir el sobreajuste del entrenamiento en redes neuronales. Esta técnica consiste en desactivar temporal y aleatoriamente diferentes unidades de la red neuronal. Esta desactivación se asocia a un parámetro  $p$  que indica la probabilidad de desactivar dicha unidad.

Esta técnica ha sido implementada en redes neuronales *feed forward* [37] y en redes neuronales recurrentes [38, 39, 40], donde la idea general sigue siendo la misma.

#### 2.3.4.4. Linear Warm-Up Schedule

El *Linear Warm-Up Schedule* es una técnica utilizada para ajustar la tasa de aprendizaje al principio del entrenamiento [41]. Esta técnica aumenta linealmente la tasa de aprendizaje desde un valor  $\alpha$  hasta un  $\beta$ . Definir un  $\alpha$  bajo permitiría reducir la volatilidad en las primeras épocas del entrenamiento. La formulación de esta técnica es la siguiente:

$$p = \alpha + (\beta - \alpha) \cdot \frac{\min(\text{epoca}, \Delta_{\text{epoca}})}{\Delta_{\text{epoca}}}, \quad (2.22)$$

donde  $p$  es el factor que multiplica a la tasa de aprendizaje,  $\alpha$  y  $\beta$  son el factor de inicio y término respectivamente de los valores que se quiere alcanzar; *epoca* corresponde a la época en la que se encuentra, y  $\Delta_{\text{epoca}}$  es el número de épocas por la que se quiere realizar la transición.

#### 2.3.4.5. Normalización por capas

La normalización por capas (más conocida como *layer normalization* [42]) es una técnica que tiene un efecto de regularización. Esta estima las estadísticas de normalización a partir de las entradas sumadas a las neuronas de las capas ocultas. Se utiliza principalmente para RNNs y *Transformers* [41].

## 2.4. Aprendizaje profundo en tareas de predicción

Las técnicas de aprendizaje de máquinas han sido fuertemente utilizadas para realizar diversas tareas por su capacidad de aprender la dinámica interna de los sistemas a partir de los datos conocidos [43]. Por otro lado, el aprendizaje profundo, un sub-tópico del aprendizaje de máquinas ha alcanzado buenos resultados en diversos campos: clasificación de imágenes [44], procesamiento natural del lenguaje [45], aprendizaje reforzado [46], entre otros.

A pesar de lo anterior, el uso de técnicas de aprendizaje profundo en tareas de predicción fue cuestionada durante un largo periodo de tiempo [47]. Esto se debía a la posibilidad de sobreajuste de los datos y a la sensibilidad del modelo a las entradas [48].

A partir de la cuarta competencia de predicción conocida como *Makridakis's competition* (M4), se mostró que los modelos de aprendizaje profundo superaron el rendimiento de los métodos más clásicos [49]. A partir de ese momento, el número de modelos predictivos que utilizan técnicas de aprendizaje aumentaron.

A continuación se formaliza y describe la tarea de predicción puntual y probabilística.

### 2.4.1. Predicciones

La tarea de predecir consiste en extrapolar las series de tiempo hacia el futuro dado su pasado [50]. Para realizar este procedimiento, se utiliza la información histórica asociada a la serie de tiempo de interés. Esta información puede ser recopilada en base a la misma serie, o a series que sean similares a ella. A continuación se define y formaliza la tarea de predicción.

### 2.4.1.1. Formalización del problema

Sea  $Z = \{z_{i,1:T_i}\}_{i=1}^N$  un conjunto de  $N$  series de tiempo con  $z_{i,1:T_i} = (z_{i,1}, \dots, z_{i,T_i})$  donde  $z_{i,t}$  es el valor de la  $i$ ésima serie de tiempo en el instante  $t$ , y  $Z_{t_1:t_2}$  corresponde a los valores de todas las  $N$  series de tiempo en el rango  $[t_1, t_2]$ . Además, sea  $X = \{X_{i,1:T_i}\}_{i=1}^N$  el conjunto de covariables donde  $x_{i,t} \in \mathbb{R}^{d_x}$ . El objetivo de la predicción es determinar la distribución condicional del futuro de la serie de tiempo, basado en la información del pasado, sus covariables y los parámetros del modelo matemático. Lo anterior se representa por la siguiente probabilidad condicional:

$$p(Z_{t+1:t+h} | Z_{1:t}, X_{1:t+h}; \theta), \quad (2.23)$$

donde  $h$  corresponde al horizonte de predicción, es decir, cuánto tiempo se mira hacia el futuro, y  $\theta$  corresponde a los parámetros del modelo que se esté utilizando. Para evitar ambigüedades se define el pasado y futuro de una serie de tiempo entre los rangos  $[1, t]$  y  $[t+1, t+h]$  respectivamente. A partir de la Ec. (2.23) se pueden definir 3 arquitecturas de modelos [50]:

**Modelo local univariado:** Un modelo local es entrenado independientemente para cada una de las  $N$  series de tiempo. La Ec. (2.23) se simplifica como sigue:

$$p(z_{t+1:t+h} | z_{1:t}, X_{1:t+h}; \theta), \quad (2.24)$$

con  $\theta_i = \psi(z_{i,1:t}, X_{i,1:t+h})$ , donde  $\psi$  es una función genérica que transforma  $z_{i,1:t}$  y  $X_{i,1:t+h}$  a un conjunto de parámetros  $\theta_i$ .

**Modelo global univariado:** Un modelo global es entrenado usando todos los datos de las  $N$  series de tiempo. La Ec. 2.23 quedaría de la siguiente forma:

$$p(z_{t+1:t+h} | Z_{1:t}, X_{1:t+h}; \theta), \quad (2.25)$$

con  $\theta_i = \psi(z_{i,1:t}, X_{i,1:t+h}, \phi)$ , donde  $\psi$  es una función genérica que transforma  $z_{i,1:t}$ ,  $X_{i,1:t+h}$  y  $\phi$  ( parámetros compartidos a lo largo de  $N$  series de tiempo), a un conjunto de parámetros  $\theta_i$ .

**Modelo multivariado:** Un solo modelo es aprendido a partir de  $N$  series de tiempo usando todos los datos conocidos, directamente prediciendo varias variables de interés. La Ec. (2.23) quedaría de la siguiente forma:

$$p(Z_{t+1:t+h} | Z_{1:t}, X_{1:t+h}; \theta), \quad (2.26)$$

con  $\theta_i = \psi(Z_{i,1:t}, X_{i,1:t+h}, \phi)$ , donde  $\psi$  es una función genérica que transforma  $Z_{i,1:t}$ ,  $X_{i,1:t+h}$  y  $\phi$  ( parámetros compartidos a lo largo de  $N$  series de tiempo), a un conjunto de parámetros  $\theta_i$ .

En los dos últimos casos,  $\phi$  corresponde a un modelo matemático global. Este suele ser una red neuronal recurrente. Además, en base a la Ec. (2.23) se encuentran dos modelos de interés: Los modelos de predicción puntual y los modelos de predicción probabilístico.

### 2.4.1.2. Modelos de predicción puntual

Los modelos de predicción puntual predicen un único valor para cada instante de tiempo de una variable de interés. Para realizarlo utilizan una función que relaciona la información

del pasado junto a sus características. Las formulaciones asociadas a este modelo son:

$$\hat{z}_{i,t+1:t+h} = \psi(z_{i,1:t}, X_{i,1:t+h}), \quad (2.27)$$

$$\hat{z}_{i,t+1:t+h} = \psi(z_{i,1:t}, X_{i,1:t+h}, \phi), \quad (2.28)$$

$$\hat{Z}_{i,t+1:t+h} = \psi(Z_{1:t}, X_{1:t+h}, \phi), \quad (2.29)$$

donde  $\hat{z}_{i,t+1:t+h}$  corresponde a las predicciones de la serie de tiempo univariada  $i$  mientras que  $z_{i,1:t}$  corresponde a la información su pasado. Por otro lado  $\psi$  es la función utilizada para predecir,  $X_{i,1:t+h}$  son las covariables conocidas en el pasado y futuro, y  $\phi$  corresponde a los parámetros asociados al modelo. Por último,  $\hat{Z}_{i,t+1:t+h}$  corresponde a las predicciones puntuales de la serie de tiempo multivariada, mientras que  $Z_{1:t}$  corresponde a la información de su pasado. Por otro lado  $X_{1:t+h}$  son las covariables de la serie de tiempo multivariada en el pasado y futuro.

La Ec. (2.27) corresponde a la arquitectura univariada local, la Ec. (2.28) corresponde a la arquitectura univariada global y la Ec. (2.29) corresponde a la arquitectura multivariada.

### 2.4.1.3. Modelos de predicción probabilística

Los modelos de predicción probabilístico modelan la distribución de probabilidad de los valores futuros de una variable de interés. Para realizarlo utilizan una función que relaciona la información del pasado junto a sus características. Este tipo de predicción es más completa que las puntuales, dado que modelan la incerteza de su predicción. Las formulaciones asociadas a los modelos de predicción probabilística son:

$$p(z_{i,t+1:t+h} | z_{i,1:t}, X_{i,1:t+h}; \theta_i) \quad \theta_i = \psi(z_{i,1:t}, X_{i,1:t+h}), \quad (2.30)$$

$$p(z_{i,t+1:t+h} | Z_{1:t}, X_{1:t+h}; \theta_i) \quad \theta_i = \psi(z_{i,1:t}, X_{i,1:t+h}, \phi), \quad (2.31)$$

$$p(Z_{t+1:t+h} | Z_{1:t}, X_{1:t+h}; \theta) \quad \theta = \psi(Z_{1:t}, X_{1:t+h}, \phi), \quad (2.32)$$

donde la definición de cada término es análoga al caso de los modelos de predicción puntual, solo considerando que en lugar de predecir un punto  $\hat{z}$ , se desea determinar la distribución  $p$ .

La Ec. (2.30) corresponde a la arquitectura univariada local, la Ec. (2.31) corresponde a la arquitectura univariada global y la Ec. (2.32) corresponde a la arquitectura multivariada.

## 2.4.2. Predicción de series de tiempo con RNNs

La literatura relacionada a la predicción de series de tiempo astronómicas es escasa (ver Sección 1.2). A pesar de lo anterior, existe una gran diversidad de técnicas que abordan el problema de predicción en otras áreas. Algunas de ellas se enfocan en utilizar *convolutional neural networks* (CNN) [51,52], *transformers* [53,54], procesos gaussianos [55] entre otros [56]. Además, existen trabajos que no solo proponen técnicas de predicción, sino que también muestran estrategias para mejorar los resultados. Por ejemplo, uno de estos trabajos propone 12 trucos para mejorar la etapa de entrenamiento y predicción de las RNNs [57]. Mientras que en [58] se proponen formas de robustecer la tarea de predicción ante entradas ruidosas.

En este trabajo se abordará una de las arquitecturas más utilizadas para la tarea de predicción, las redes neuronales recurrentes. Estas permiten modelar la información temporal de datos secuenciales como lo son las series de tiempo. A partir de esta característica, es que este tipo de red naturalmente se utiliza para predecir. Dependiendo lo que se estime, las RNNs pueden obtener predicciones puntuales o probabilísticas.

Los modelos de predicción puntual se caracterizan por entregar una única predicción de los instantes de tiempo que se quiera predecir. Este tipo de modelo, a pesar de abundar en la literatura [57, 59, 60, 61, 62, 63], suelen estar incompletos al no entregar un grado de incertidumbre de sus predicciones. Es por esto, que se han propuesto formas para transformar los modelos puntuales a modelos probabilísticos en base a la modificación o perturbación de los datos entrantes, funciones de costo, etc [50, 64].

Por otro lado, están los modelos probabilísticos que utilizan RNNs para estimar la distribución de probabilidad de los instantes de tiempo futuro que se quiera predecir. Al igual que para el caso de las predicciones puntuales, estos modelos abundan en la literatura, donde se puede encontrar modelos univariados [62, 65, 66, 67] y multivariados [68, 69].

Usualmente la tarea de estimar la distribución de probabilidad suele ser intratable o costosa computacionalmente, por lo que en general se suele estimar a partir del muestreo de Monte Carlo [70]. Para realizar lo anterior, los modelos parametrizan distribuciones probabilísticas conocidas [63, 67, 71, 72, 73, 74] con las cuales luego se genera el respectivo muestreo.

El modelo propuesto busca explorar esta estrategia y adaptarla para trabajar sobre curvas de luz de supernovas. En particular, el modelo utilizado como inspiración corresponde al propuesto en [74]. Este modelo propone predecir en base a una técnica de aprendizaje profundo que utiliza redes neuronales recurrentes para parametrizar funciones probabilísticas. Dado lo anterior, este modelo es global univariado pues aprende patrones en base a un conjunto relacionado de series de tiempo.

Las entradas del modelo corresponden a una combinación de las covariables asociadas a los instantes de tiempo pasados y futuros. Además, una de las entradas más importantes corresponde al valor predicho en el instante anterior, el cuál entra de forma recursiva. Este modelo luego de aprender el contexto de las observaciones conocidas, puede generar predicciones multihorizontes en base a una metodología iterativa. Lo descrito anteriormente se detalla en la Sección 3.

#### 2.4.2.1. Características o covariables

En general, los modelos de aprendizaje profundo suelen verse afectados según la calidad y cantidad de información entregada. En particular, dado que las RNNs pueden recibir información secuencial, también reciben información asociada al tiempo. A partir de esta característica, el modelo sería capaz de generar relaciones temporales que permitan determinar la dinámica del sistema [75].

Las características (o covariables) pueden ser dependientes o independientes del tiempo [65, 71, 74] y pueden poseer incertezas [72]. Dicho lo anterior, a continuación se muestra una serie de características comúnmente utilizadas en diversas áreas de investigación:

- **Edad:** Distancia de tiempo que existe entre el primer punto y el resto de una misma

serie de tiempo [11, 74]

- **Categoría** : Característica independiente del tiempo. Se suele asociar a la categoría o clasificación de la serie de tiempo en estudio. En este caso, la categoría tiene que ver con el tipo de supernova y la banda en estudio.
- **Diferencia de tiempo entre puntos** ( $\Delta t$ ) : Diferencia de tiempo entre puntos. [59, 76, 77, 78, 79, 80, 81].
- **Discretización**: Cuando una serie de tiempo es continua, esta puede ser discretizada en un conjunto finito de valores, lo cual permite entregar otro grado de interpretación a los datos [82]. Una típica discretización temporal corresponde al *binning* donde los datos se acomodan dentro de *buckets* que a su vez sirven como entradas para el modelo.
- **Máscara**: Indicador de alguna característica en particular del dato. Generalmente se utiliza para indicar si el dato existe o no, si es real o una proyección [83]. También puede ser utilizada para indicar el momento de eventos de interés en una serie de tiempo.
- **Valores pasados**: Pasado de la variable a predecir [71, 74].
- **Codificadores temporales**: Codificación del tiempo mediante funciones conocidas. Por ejemplo puede ser a través de funciones senos y cosenos [41].
- **Porcentaje de valores perdidos**: Porcentaje de valores faltantes en una serie de tiempo. Esta característica es útil cuando se quiere estudiar el proceso de medición u observación de la característica de interés [83].
- **Error**: Información asociada a la incerteza de la medición u observación. En astronomía este dato suele ser calculado cuando se obtiene el flujo fotométrico.
- **Diferencia de magnitud**: Diferencia de magnitud entre los valores de interés. Esta característica permite conocer cuánto varía la variable según el paso del tiempo [7, 57]

### 2.4.3. Irregularidad y pérdida de valores

Los problemas de irregularidad en el muestreo de los datos y la pérdida de valores suelen estar relacionados entre sí. Una forma de unificar ambos problemas, sería asumiendo que las series de tiempo irregularmente muestreadas corresponden a series de tiempo regularmente muestreadas pero con la presencia de datos faltantes [5].

Este tipo de problemas suele ser perjudicial a la hora de analizar series de tiempo, pues se pierde continuidad de las mediciones. Además, los modelos de aprendizaje profundo como las RNNs suelen empeorar su rendimiento si los datos entrantes están irregularmente muestreados [27]. Dado lo anterior, dos estrategias comunes para solucionar este problema son (1) usar estrategias de imputación de datos o (2) desarrollar modelos que pueden manejar directamente los datos. Ambas se describen a continuación.

#### 2.4.3.1. Imputación

La imputación corresponde a reemplazar u agregar un valor en el instante de tiempo en el que se sabe que falta una medición. Esto se puede realizar mediante técnicas de reemplazo [84],



interpolación o a través del uso de técnicas de aprendizaje de máquinas [85]. Algunos ejemplos se observan en [86] donde se propone la utilización de dos técnicas: la de imputación hacia adelante, que consiste en reemplazar el valor perdido por el último valor conocido; o el relleno por modelo, que consiste en utilizar un modelo matemático, como la regresión lineal, para reemplazar el valor faltante. También en [87] se propone la utilización de una red que genera interpolaciones a partir de serie de tiempo irregularmente muestreadas. Otra alternativa, sería la de generar la imputación directamente en base a la salida del modelo [27].

### 2.4.3.2. Modelos

La estrategia relacionada al modelo, consiste en generar y/o modificar alguna arquitectura que pueda trabajar directamente con los datos irregulares, realizando preprocesamientos menores.

En la literatura, existe una diversidad de técnicas que modifican la arquitectura de modelos ya conocidos. En [61] proponen una nueva clase de RNN que fue diseñada para compensar los valores faltantes. Por otro lado, en [83] se modifica el bloque de una RNN para agregar componentes de decaimiento para interpretar la irregularidad temporal de los datos. Además en este último artículo se propone la utilización de un indicador para determinar cuando se trata de una observación conocida y cuando de un valor perdido, aumentando así el conocimiento del modelo. En otros trabajos, se propone la utilización de modelos autoregresivos basados en la representación discreta de modelos autoregresivos continuos para trabajar sobre curvas de luz [88, 89].

## 2.5. Método Monte Carlo

El método de Monte Carlo es una técnica estadística utilizada para estimar los posibles resultados de un suceso incierto [90]. Este método se basa en simular distintos escenarios y generar un gran número de resultados aleatorios. A partir de estos resultados se puede obtener una distribución de probabilidades del suceso en cuestión.

En el contexto de las predicciones, el método de Monte Carlo es utilizado para generar predicciones probabilísticas. Estas predicciones se generan a partir de la simulación de muchos escenarios diferentes (basándose en el pasado y características de la serie de tiempo que busca ser predicha). Cada simulación generará una posible predicción. A partir de la combinación de todas predicciones, se obtiene una distribución de probabilidad de las predicciones.

## 2.6. Cadenas de Markov

Una cadena de Markov es un proceso estocástico que describe la evolución dinámica de un sistema en el que la probabilidad de transición de un estado discreto solo dependerá del estado actual [91]. Formalmente, una cadena de Markov  $X_1, \dots, X_n$  es una secuencia de variables aleatorias (no i.i.d) que cumplen con la propiedad de Markov. Esta última se define como:

$$p(X_{n+1}|X_n, X_{n-1}, \dots, X_0) = p(X_{n+1}|X_n), \quad (2.33)$$

es decir, que la probabilidad de que ocurra un evento  $X_{n+1}$  dependerá solo del estado  $X_n$ .

## 2.7. Markov Chain Monte Carlo

El *Markov Chain Monte Carlo* (MCMC), es una estrategia de inferencia de Monte Carlo que se basa en el uso de cadenas de Markov [92]. Esta estrategia es utilizada para estimar distribuciones cuando son muy complejas (o intratables) de obtener por métodos analíticos [93]. El objetivo de los algoritmos MCMC es extraer  $M$  muestras de la densidad de probabilidad posterior.

$$p(\Theta, \alpha|D) = \frac{p(\Theta, \alpha)p(D|\Theta, \alpha)}{p(D)}, \quad (2.34)$$

donde  $\Theta$  es el conjunto de parámetros del modelo,  $\alpha$  es un conjunto adicional de parámetros,  $p(\Theta, \alpha)$  es la distribución a priori y  $p(D|\Theta, \alpha)$  es la función de verosimilitud que puede ser obtenida computacionalmente para cualquier valor de  $(\Theta_i, \alpha_i)$ .  $p(D)$  es la constante de normalización que es independiente de  $\Theta, \alpha$  una vez escogido la forma del modelo generativo. Lo anterior implica que es posible conocer la distribución posterior sin conocer  $p(D)$ .

Una vez realizadas las simulaciones, es posible estimar el valor esperado de una función de los parámetros del modelo:

$$E(f(\Theta)) = \int p(\Theta|D)f(\Theta)d\Theta \approx \frac{1}{M} \sum_{i=1}^M f(\Theta_i). \quad (2.35)$$

El MCMC realiza una caminata aleatoria (*random walk*) sobre el espacio de parámetros con el objetivo de obtener un conjunto representativo de muestras de la distribución. El procedimiento anterior se realiza a través de una cadena de Markov, es decir, cada uno de los estados  $X_i = [\Theta_i, \alpha_i]$ , dependerá solo del estado anterior  $X_{i-1}$ .

En el presente trabajo de tesis se trabaja con la librería *emcee* [93], que implementa la técnica de *affine invariant ensemble sampler* para el uso de la estrategia de MCMC. Lo anterior fue propuesto en [94].

## 2.8. Métricas de evaluación

En la literatura, existe una gran cantidad de métricas de evaluación relacionadas a la predicción. Según el tipo de predicción se puede clasificar en métricas puntuales y probabilísticas. Además, según la dependencia a la escala se pueden clasificar en dependiente e independiente. En esta sección se mostrarán los distintos tipos de métricas utilizadas en este trabajo.

### 2.8.1. Métricas puntuales

Las métricas puntuales comparan el valor predicho con el valor deseado. A continuación se muestran las métricas utilizadas en este trabajo.

### 2.8.1.1. Error medio absoluto

El error medio absoluto (MAE, por sus siglas en inglés) es una métrica dependiente de escala que tiene la siguiente forma:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.36)$$

donde  $y_i$  es el valor deseado,  $\hat{y}_i$  es el valor predicho y  $n$  es la cantidad de datos.

### 2.8.1.2. Error medio absoluto escalado

El error medio absoluto escalado (MASE, por sus siglas en inglés; [95]), corresponde a la versión escalada del MAE. Esta métrica es independiente de escala y tiene la siguiente forma:

$$MASE = \frac{1}{n} \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - y_{naive}|}, \quad (2.37)$$

donde  $y_{naive}$  es la predicción de un modelo más simple,  $y_i$  es el valor deseado,  $\hat{y}_i$  es el valor predicho y  $n$  es la cantidad de datos.

Esta métrica es muy utilizada en la literatura [49, 60, 63, 96] por su interpretabilidad. Al tratarse de una comparación entre el modelo predicho y un modelo más simple. Si el MASE es inferior a 1 significa que el modelo es mejor que un modelo *naive*. En caso contrario (MASE > 1), el modelo *naive* sería mejor. Esta métrica tiene la desventaja de que puede generar valores infinitos o indefinidos cuando el denominador es muy pequeño.

### 2.8.1.3. Error porcentual medio absoluto

El error porcentual medio absoluto (MAPE, por sus siglas en inglés) es una métrica independiente de escala que mide el error en términos porcentuales. Tiene la siguiente forma:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right|, \quad (2.38)$$

donde  $y_i$  es el valor deseado,  $\hat{y}_i$  es el valor predicho y  $n$  es la cantidad de datos.

Esta métrica es muy utilizada en la literatura [55, 61, 68, 75] por su fácil interpretación. A pesar de esto, al igual que MASE, puede indefinirse cuando el denominador es muy pequeño. Además, tiene una mayor penalización en los errores positivos en comparación a los negativos [95]. Sus valores varían entre 0 y 100 %

### 2.8.1.4. Error porcentual medio absoluto simétrico

El error porcentual medio absoluto simétrico (sMAPE, por sus siglas en inglés; [97]) es una modificación del MAPE. Tiene la forma:

$$sMAPE = \frac{200}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}, \quad (2.39)$$

donde  $y_{naive}$  es la predicción de un modelo más simple,  $y_i$  es el valor deseado,  $\hat{y}_i$  es el valor predicho y  $n$  es la cantidad de datos.

Esta métrica es muy utilizada en la literatura [49,56,60,63,68]. Busca resolver el problema de los errores indefinidos. Sus valores varían entre 0 y 200 %.

## 2.8.2. Métricas probabilísticas

Las métricas probabilísticas comparan el intervalo de predicción y/o la distribución de los valores predichos con el valor deseado. Los dos criterios de evaluación críticos corresponden a *confidence* y *sharpness* [98,99]. La *confidence* se encarga de asegurar que no existe sesgo en las predicciones. Por otro lado el *sharpness* corresponde a la eficiencia relacionada a la amplitud de los intervalos de predicción.

### 2.8.2.1. Probabilidad de cobertura del intervalo de predicción

La probabilidad de cobertura de intervalos de predicción (PICP; por sus siglas en inglés) es una métrica que estudia si los valores deseados se encuentran dentro de un intervalo de predicción. Las ecuaciones que rigen esta métrica corresponden a:

$$\beta_i = \begin{cases} 1 & \text{si } z_i \in [L_i, U_i] \\ 0 & \text{si } z_i \notin [L_i, U_i] \end{cases}, \quad (2.40)$$

donde  $L_i$  corresponde al límite inferior y  $U_i$  al límite superior del intervalo de predicción.

Dado lo anterior, se obtendrá el promedio de puntos que caen o no, dentro del intervalo de predicción a partir de:

$$PICP = \frac{1}{n} \sum_{i=1}^n \beta_i, \quad (2.41)$$

donde  $n$  corresponde al número de datos.

La utilización de esta métrica no es tan masiva como las anteriores [100]. A pesar de lo anterior, esta permite robustecer los análisis al momento de comparar entre los distintos modelos probabilísticos.

### 2.8.2.2. Puntuación media del intervalo

La puntuación media del intervalo (MSIS, por sus siglas en inglés) corresponde a la versión escalada del *Winkler Score* [70,101] la cual se utiliza para medir el *confidence* de un modelo. La forma de la ecuación es la siguiente:

$$MSIS = \frac{\frac{1}{n} \sum_{i=1}^n (U_i - L_i) + \frac{2}{a}(L_i - y_i)\mathbf{1}\{y_i < L_i\} + \frac{2}{a}(y_i - U_i)\mathbf{1}\{y_i > U_i\}}{\frac{1}{n} \sum_{i=1}^n |y_i - y_{naive}|}, \quad (2.42)$$

donde  $U_i$  es el límite superior y  $L_i$  es el límite inferior del intervalo de predicción;  $\mathbf{1}$  es el indicador de si el valor está por sobre o por debajo de los límites,  $y_{naive}$  es la predicción de

un modelo más simple,  $y_i$  es el valor deseado,  $n$  es la cantidad de datos y  $a$  es el nivel de significancia.

Esta métrica, en su versión no escalada, es muy utilizada [63, 66, 99], mientras que su versión escalada no es tan recurrente [49].

### 2.8.2.3. Función de pérdida cuantílica escalada

La función de pérdida cuantílica escalada, también conocida como *Scaled Pinball Loss* (SPL) corresponde a la versión escalada del *Pinball Loss* (PL). Esta métrica evalúa la precisión de la predicción probabilística a partir de la siguiente ecuación:

$$SPL(u) = \frac{1}{n} \frac{\sum_{t=1}^n (y_i - Q_i(u))u \mathbf{1} + (y_i - Q_i(u))u \mathbf{1} + (Q_i(u) - y_i)(1 - u) \mathbf{1}\{Q_i(u) < y_i\}}{\frac{1}{n} \sum_{i=1}^n |y_t - y_{naive}|}, \quad (2.43)$$

donde  $Q_i(u)$  es el valor del percentil  $u$ ;  $\mathbf{1}$  es el indicador de si el valor está por sobre o por debajo del percentil,  $y_{naive}$  es la predicción de un modelo más simple,  $y_i$  es el valor deseado, y  $n$  es la cantidad de datos.

El PL ha sido muy utilizado en la literatura [54, 63, 65, 66, 67, 73, 99]. Por otro lado, su versión escalada no es tan utilizada, pero es útil para que la métrica no dependa de la escala [75, 102].

### 2.8.2.4. Puntuación continua de probabilidad escalada

La puntuación continua de probabilidad (CRPS, por su siglas en inglés; [70]) es una de las métricas más versátiles para evaluar predicciones probabilísticas. Su ecuación es la siguiente:

$$CRPS(F, y) = \int_{-\infty}^{\infty} (F(\hat{y}) - \mathbf{1}\{\hat{y} \geq y\})^2 d\hat{y}, \quad (2.44)$$

donde  $y$  es el valor deseado,  $\hat{y}$  son las predicciones asociadas a un instante y  $F$  es la función de distribución acumulada, además

$$\mathbf{1} = \begin{cases} 0 & \text{si } y < \hat{y} \\ 1 & \text{si } y \geq \hat{y} \end{cases}. \quad (2.45)$$

Debido a que no se conoce la función de distribución acumulada, en [103] se propone aproximar el CRPS a partir de la siguiente ecuación:

$$CRPS(y) = \frac{1}{n} \sum_{i=1}^n |X_i - y| - \frac{1}{2n^2} \sum_{i=1}^n \sum_{j=1}^n |X_i - X_j|, \quad (2.46)$$

donde  $y$  es el valor deseado,  $X_i$  a todas las predicciones asociadas al instante  $i$  y  $n$  es el número de instantes de tiempo.

Esta métrica es muy utilizada en la literatura [55, 63, 67, 68, 69, 72, 99, 104, 105] por su versatilidad. Además, esta corresponde a una generalización del MAE lo cual permite tener una intuición de sus resultados.

A pesar de que en la literatura no se encontró la versión escalada de esta métrica, se seguirá la misma idea utilizada en el MSIS y SPL, por lo que la versión escalada del CRPS tiene la siguiente forma:

$$CRPSS(y) = \frac{CRPS(y)}{\frac{1}{n} \sum_{i=1}^n |y_t - y_{naive}|}. \quad (2.47)$$

## 3. Metodología

En esta sección se describe la metodología utilizada para abordar el problema de predicción de curvas de luz de SNe. Para empezar, se presenta la base de datos a utilizar junto a los respectivos preprocesamientos. Luego, aprovechando la información del error de observación y al modelo paramétrico, se proponen técnicas para aumentar la cantidad de datos. A continuación se presenta la estrategia utilizada para reducir la irregularidad en el muestreo de los datos. Se explican las estrategias de entrenamiento y de predicción del modelo propuesto.

### 3.1. Datos y preprocesamiento

La base de datos fue provista por el equipo de ALeRCE - ZTF y corresponde a la versión 8.0.3. Esta base de datos está compuesta por curvas de luz de 4 clases de supernovas: SNIa, SNIbc, SNII (incluye SNIIn) y SLSN (*Superluminous SNe*). Además, cada supernova tiene la información asociada a dos bandas fotométricas: banda  $g$  y banda  $r$ .

En la Fig. 3.1 se observa la distribución de las muestras existentes para cada uno de los tipos de supernova. Se observa el claro desbalance entre clases destacando que la clase SNIa corresponde a la clase mayoritaria y la SLSN a la minoritaria.

Dado que en un principio se busca realizar predicciones univariadas, se trabajará con la información de cada banda de forma separada. Dado lo anterior, la curva de luz de supernova será dividida en sus dos bandas, las cuales serán tratadas como dos curvas de luz independientes.

Para que una curva de luz (de una sola banda) sea considerada como válida, es decir, una curva que pueda ser utilizada para entrenar, evaluar y probar, se realiza una etapa de preprocesamiento. A continuación se presenta una lista con todos los procesos realizados:

- **Número mínimo de puntos:** Cada curva de luz debe poseer una cantidad mínima de 4 puntos. Este filtro asegura que exista suficiente información para utilizar modelos que ajustan los datos a modelos matemáticos. Estos ajustes serán aprovechados en el entrenamiento, aumento de datos y evaluación del modelo.
- **Máxima cantidad de días:** Solo se considerarán los puntos con una lejanía de máximo 100 días con respecto a la primera observación. Este filtro permite reducir el costo computacional del modelo, sin perder la dinámica relevante de las supernovas.
- **Máximo flujo:** Solo se considerarán las curvas de luz que no posean un flujo normalizado mayor a 1. Este filtro permite estabilizar los errores procesados en la etapa de

### Distribución de clases ZTF v8.03

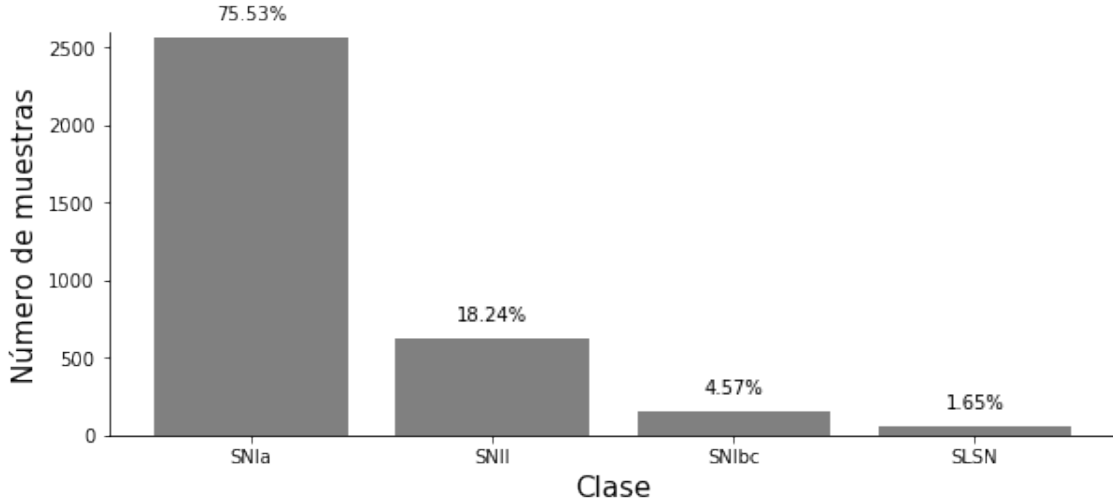


Figura 3.1: Gráfico de barra que compara el número de supernovas por tipo (SNIa, SNIi, SNIbc y SLSN) presentes en la base de datos provista por ALeRCE - ZTF en su versión 8.0.3.

*backpropagation*.

- **Una observación por día:** En este trabajo se plantea una metodología que requiere que solo se considere una observación por día. Dado lo anterior, si existen dos observaciones en el mismo día, se escoge aleatoriamente uno de estos valores.
- ***sigma clipping*:** Una propuesta habitual es realizar una eliminación de muestras en base al error de observación [19]. Si el error de observación es superior a 5 veces la mediana de errores del conjunto de entrenamiento de un determinado *kfold*, entonces esta muestra es eliminada.

Según lo anterior, todas las curvas de luz que sigan teniendo más de 4 observaciones serán consideradas como supernovas válidas.

#### 3.1.1. Características por tipo de supernova

En este trabajo de tesis se realiza un análisis diferenciado para cada uno de los tipos de supernova. Dado lo anterior, a continuación se presenta una descripción de los datos asociados a cada uno de los tipos de supernova.

En la Tabla 3.1 se observa la cantidad de muestras válidas por banda de cada tipo de supernova. Como se observa, existe un gran balance entre curvas de tipo g y de tipo r, por lo que no se realizará ninguna técnica de balanceo asociada a las bandas.

Por otro lado en la Fig. 3.2 se muestra la distribución de tiempo que tarda cada tipo de curva de luz en llegar a su valor máximo desde su primera observación. Para aproximar el momento de máximo brillo se comparó la distancia entre la primera observación y el punto



Tabla 3.1: Número de curvas válidas por tipo de supernova para cada una de las bandas

Tipo de SNe	Número de curvas de luz válidas banda g	Número de curvas de luz válidas banda r
SNIa	2439	2443
SNI	597	593
SNIbc	146	145
SLSN	54	54

**observado** de máximo brillo. Dado lo anterior se define una forma de clasificar cada curva de luz según su brillo:

- Detecciones tardías (después del máximo brillo): Significa que la detección de la supernova se hizo justo o después del momento del máximo brillo. Todas las curvas de luz cuya primera observación posea el brillo más alto se considera como detección tardía.
- Detecciones tempranas (antes del máximo brillo): Significa que la detección de la supernova se hizo antes del momento del máximo brillo. Todos los elementos cuya primera observación es menor que las observaciones que le siguen, se considera como detección temprana.

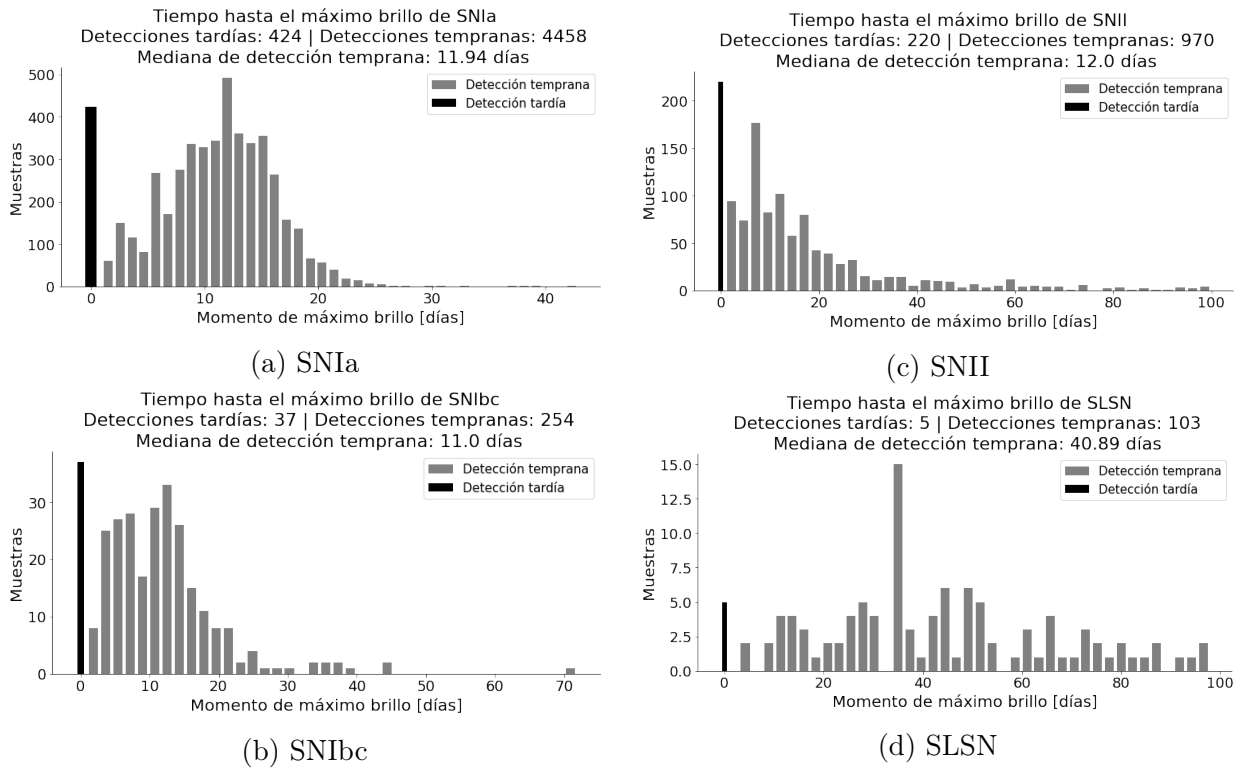


Figura 3.2: Histogramas por cada tipo de supernova que muestra la cantidad de muestras detectadas tardíamente y tempranamente. Además, se agrega información sobre la mediana de detección temprana.

Otro elemento de interés corresponde a la diferencia de tiempo entre observaciones. La distribución de las diferencias de tiempo se muestran en la Fig. 3.3. Dado que esta es una

característica que depende del *survey* y no de la clase de supernova, es que se presenta un solo gráfico que resume la cadencia de las observaciones.

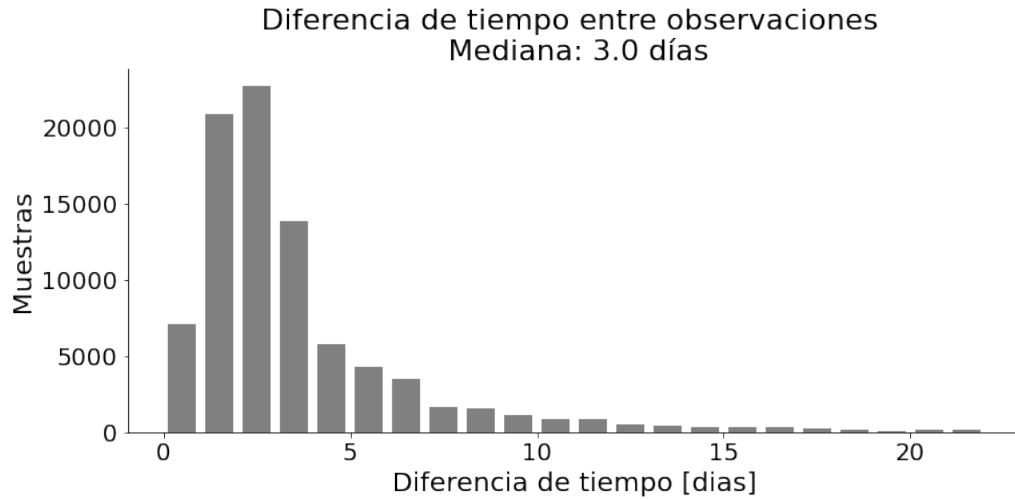


Figura 3.3: Distribución de diferencia de tiempo entre observaciones de todas las supernovas, luego de aplicar el preprocesamiento de los datos. La mediana de 3.0 días corresponde a la cadencia de ZTF.

Finalmente, en la Fig. 3.4 se muestra la distribución de número de puntos que presenta cada tipo de supernova.

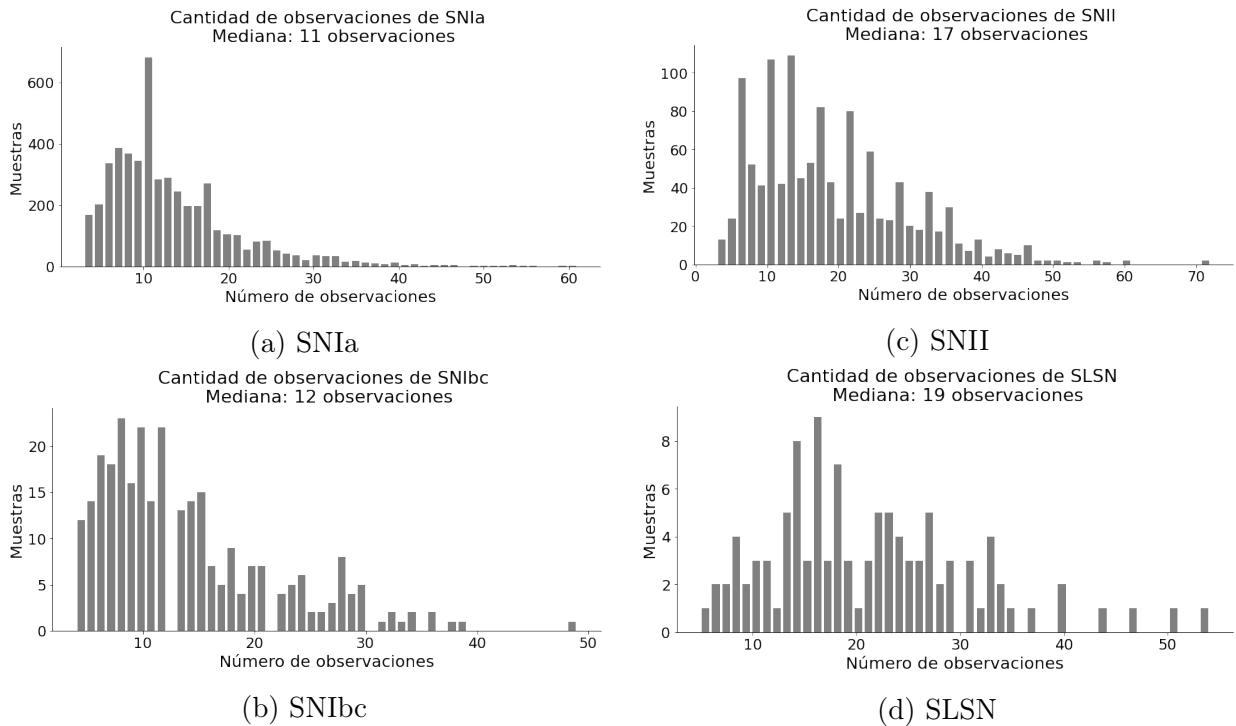


Figura 3.4: Histogramas por cada tipo de supernova que muestran la cantidad de observaciones por curva de luz. Además, se agrega información sobre la mediana de observaciones.

## 3.2. Aumento de datos

El aumento de datos es una técnica muy utilizada en aprendizaje profundo para generar más datos de entrenamiento en base a los ya conocidos. Esta técnica busca diversificar los datos para que el modelo explore un espacio paramétrico más amplio [96]. Para el caso de series de tiempo existe una gran variedad de técnicas. Una gran referencia para el estudio de cada una de estas técnicas se encuentra en [106].

En esta sección se presentan dos técnicas para aumentar la cantidad de datos basados en los datos reales que se tienen. La primera está inspirada en [19] donde se realiza un aumento de datos a partir del ajuste de los datos al modelo paramétrico de supernovas (SPM) utilizando la técnica MCMC. La segunda se inspira en [106] donde se utilizan técnicas de perturbación, recorte y eliminación de datos. Ambas propuestas son descritas a continuación.

### 3.2.1. Generación de curvas sintéticas

La primera metodología corresponde a la creación de datos sintéticos. Este aumento se realiza ajustando los parámetros del SPM mediante la técnica MCMC. Esta técnica recibe como entrada la información asociada a cada observación, las restricciones de los parámetros y a la estimación a priori de estos valores. Tanto las restricciones como la estimación de los valores se muestran en [14, 19].

A partir de la técnica MCMC es posible encontrar una diversidad de conjuntos de parámetros  $\theta_i$  que se ajustan al SPM y cumplen con las restricciones previamente otorgadas. Lo anterior permite que en base a una curva de luz dada, se generen versiones similares (pero no iguales) de a la original. Esto permite aumentar la diversidad de los datos.

Para diversificar aún más cada una de las curvas de luz sintéticas, se propone:

- Realizar muestreo irregular de los datos: es decir, cada curva de luz sintética generada tendrá una distinta cantidad de puntos y largo. Esto permite diversificar la cadencia de los datos observados logrando así que el predictor sea robusto ante cambios de cadencia.
- Perturbación en base al error de observación: dado que las curvas de luz reales presentan un error de observación, se busca replicar esto en base a asignar un error a cada observación sintética. Para obtener este error, se obtiene la distribución de errores asociados al conjunto de entrenamiento. Luego, se determina que tipo de distribución es para luego generar un muestreo aleatorio en base a esta. En la Fig. 3.5 se observa un ejemplo de la distribución obtenida en base a un conjunto de datos dado. En base a la distribución encontrada, se le asigna un error a cada una de las observaciones sintéticas, las cuales serán perturbadas. Esta perturbación se realiza siguiendo una distribución normal cuya media es la observación y la desviación estándar corresponde al error de observación.
- **Sigma clipping**: Se eliminan todas las observaciones cuyo error sea superior a 5 veces la mediana del error del conjunto de entrenamiento. Esto se realiza con el objetivo de no entregar datos erróneos al modelo.

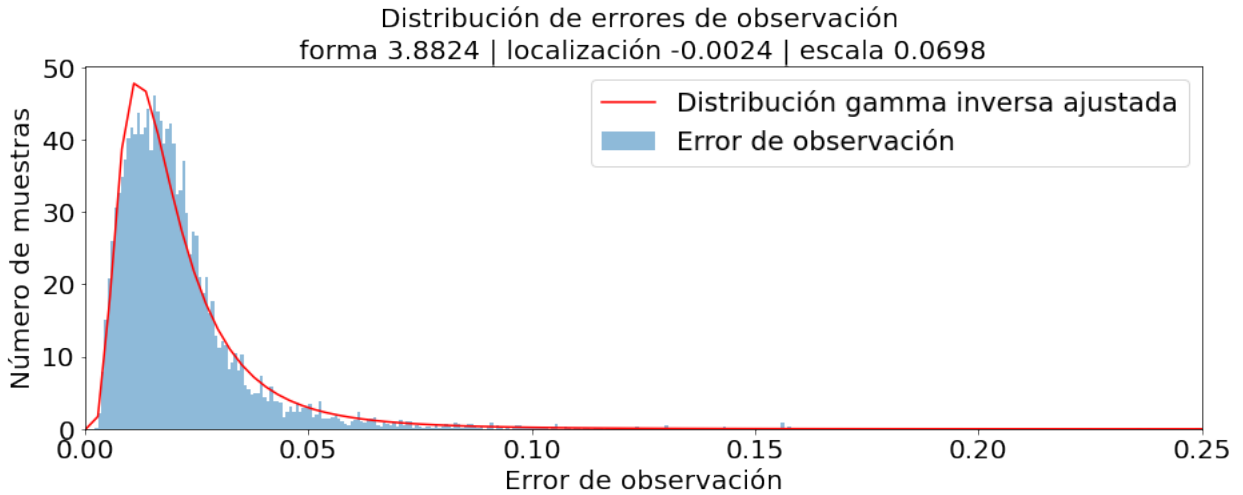


Figura 3.5: Distribución de errores de observación del conjunto de entrenamiento para una partición de los datos. En esta se observa tanto el error de observación como el ajuste de su distribución. Además se presentan los valores asociados a dicha distribución. Finalmente, a partir de esta distribución se asignarán los errores de observación a cada uno de las observaciones sintéticas.

En la Fig. 3.6 se muestra un ejemplo de aumento sintético, donde se observa una curva de luz real junto a las distintas curvas sintéticas que se obtienen de esta. Como se observa, no todas las curvas de luz inician desde el brillo más bajo, sino que algunas (como la variación 4), inician luego del momento máximo de brillo.

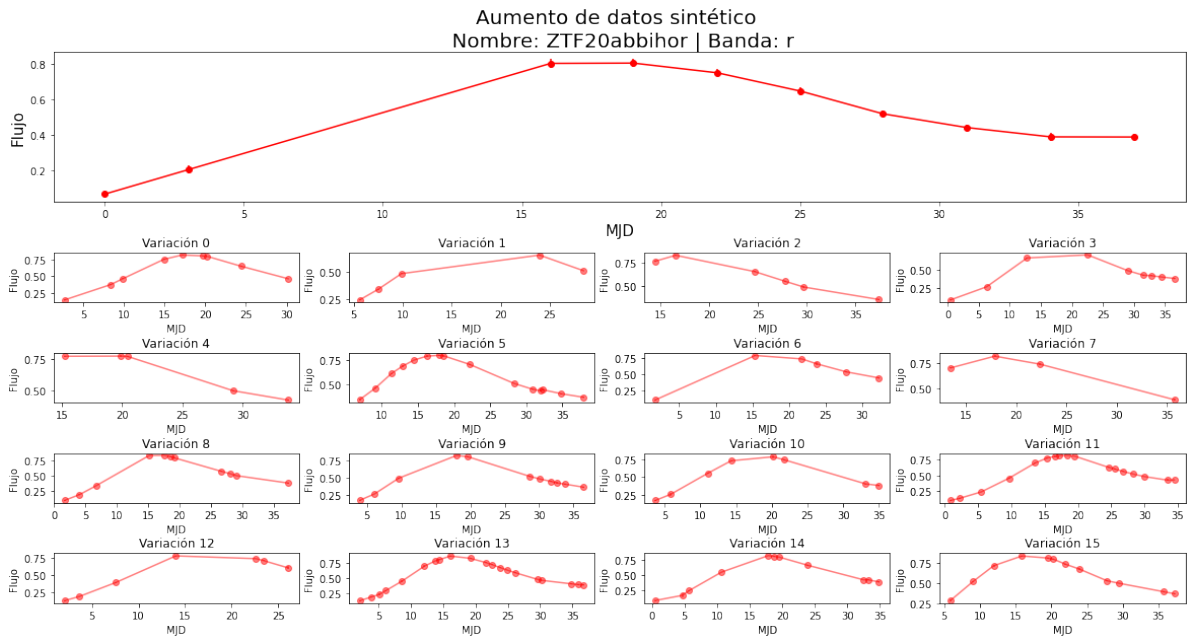


Figura 3.6: Ejemplo de aumento de datos sintéticos de la curva de luz ZTF20abbihor de la banda r. La figura de arriba corresponde a la curva de luz observada, mientras que las 16 de abajo corresponden a variaciones obtenidas en base a la estrategia propuesta.

### 3.2.2. Perturbación de curvas reales

Esta estrategia aumenta la cantidad de datos en base a modificaciones efectuadas sobre el conjunto de curvas reales. La gran ventaja de esta estrategia, es que no requiere el uso de técnicas complejas ni supuestos fuertes.

Se crearán tantas variaciones como se requieran en base a cada una de las curvas de luz del conjunto de datos de entrenamiento. Cada variación pasa por dos etapas. La primera etapa corresponde a la perturbación de las observaciones en base al error de observación. La segunda etapa busca eliminar (o no) observaciones. A continuación se detalla cada etapa.

Como se mencionó, en la primera etapa se perturban las observaciones en base al error de observación. Para realizar lo anterior, se asume que cada observación sigue una distribución normal, donde la media corresponde a la observación y la desviación estándar corresponde al error de observación. En base a esto, la curva de luz puede pasar por una de las siguientes 3 estrategias:

- **Curvas reales:** No se realiza ningún tipo de modificación. Este proceso se utiliza para que el modelo tenga oportunidad de ver la curva real. Esto permitiría conservar la distribución de las diferencias de tiempo de las observaciones reales.
- **Curvas con algunas observaciones perturbadas:** En este proceso se define la probabilidad de que el punto sea observado. En particular, se utiliza un valor del 50%. Esta modificación es útil para agregarle una variación controlada a las observaciones.
- **Curvas con todas las observaciones perturbadas:** En este proceso todos los puntos serán perturbados en base al error de observación. Este procedimiento permite diversificar el conjunto de datos existentes.

Las curvas de luz resultantes de la etapa anterior pasan a la segunda etapa. Esta consiste en diversificar aún más los datos en base a una de las siguientes 3 estrategias:

- **Conservación de los datos:** La curva generada anteriormente no presenta modificaciones. Esta estrategia daría la posibilidad de que el modelo observe la curva de luz real en caso de que en la etapa anterior la estrategia seguida fuera la de curvas reales.
- **Eliminación de algunas observaciones:** Con el fin de aumentar la diversidad en la diferencia de tiempo entre puntos, se eliminan los puntos con una probabilidad fija. En este caso existe un 30% de probabilidades que se elimine alguna observación.
- **Segmentación de la curva:** Con el fin de conservar la cadencia, se recorta parte de la curva de luz. Este procedimiento permite enfatizar zonas de interés, como lo son las primeras observaciones. Para asegurar esto, los recortes se concentran más en la inclusión de los primeros puntos y exclusión de los últimos (dado que los primeros suelen tener el punto más alto de brillo y que los últimos suelen ser zonas de poco interés por su bajo brillo y predecible comportamiento).

Un resumen de las etapas anteriores se observa en la Fig. 3.7. En esta se muestran las posibilidades de variaciones que puede sufrir cada curva de luz.

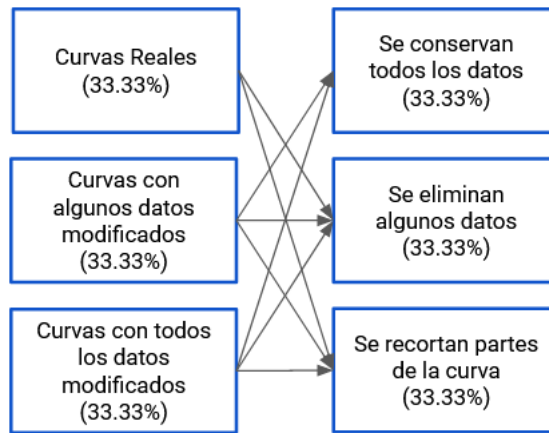


Figura 3.7: Esquema de aumento de datos sintéticos. A la izquierda se observan las estrategias de la etapa 1 y a la derecha las de la etapa 2. Cada línea representa el camino de variación que puede seguir cada curva de luz.

En la Fig 3.8 se observa un ejemplo de aumento real. En ésta se observa una curva de luz real junto a las distintas variaciones que se obtienen de esta. Como se observa, las curvas tienen distinto largo, forma y número de puntos.

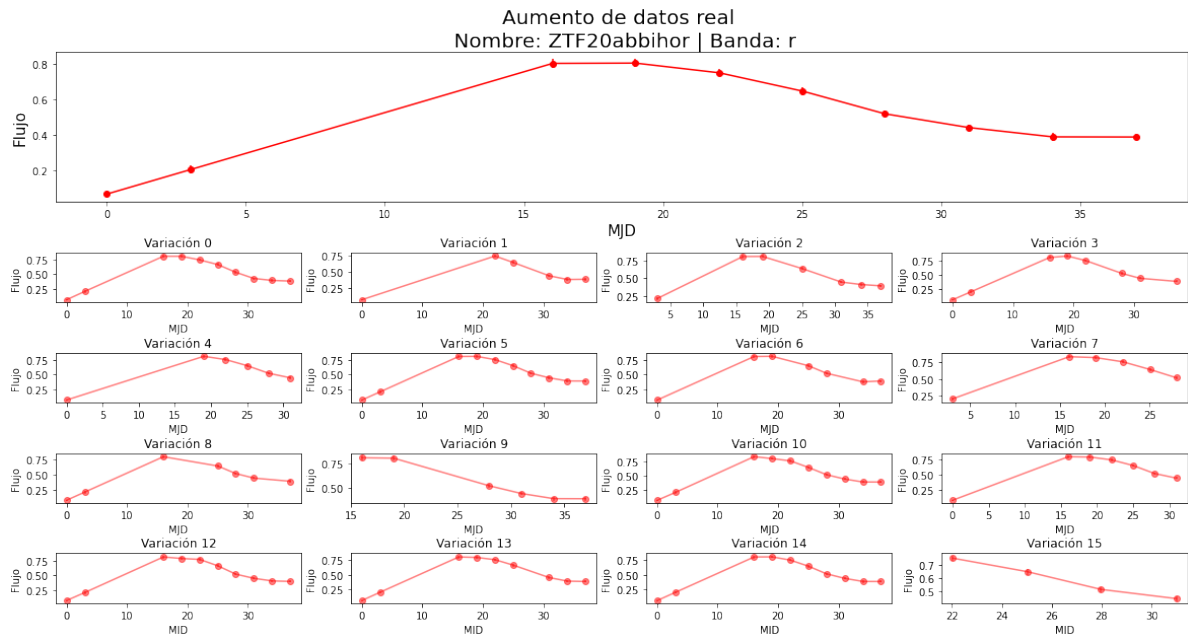


Figura 3.8: Ejemplo de aumento de datos sintéticos de la curva de luz ZTF20aabbihor de la banda r. La figura de arriba corresponde a la observación real, mientras que las 16 de abajo corresponden a variaciones obtenidas en base a la estrategia propuesta.

### 3.3. Balanceo de clases

Como fue mencionado anteriormente, el conjunto de datos con el que se trabaja presenta un gran desbalance en el número de muestras de cada tipo de supernova (Ver Fig. 3.1). Este desbalance provoca que el proceso de aprendizaje, validación y prueba se vean negativamente afectados. Para las etapas de entrenamiento y evaluación, el modelo tenderá a sobreajustarse a la clase mayoritaria, evitando así que aprenda el comportamiento de las clases minoritarias. Además, para la etapa de prueba, afectará al resultado de las métricas pudiendo provocar que estas no sean representativas.

Debido a que cada una de las etapas (entrenar, validar y probar) tiene sus propias complicaciones asociadas, estas serán abordadas de forma separada. A continuación se describe el proceso utilizado para disminuir el desbalance de datos.

#### 3.3.1. Conjunto de entrenamiento

Abordar el problema de desbalance del conjunto de entrenamiento es de vital importancia para que el modelo aprenda el comportamiento de cada tipo de supernova y no se sobreajuste al de la clase mayoritaria.

Dado lo anterior, se propone una metodología para equiparar la cantidad de datos minoritarios a la clase mayoritaria. A continuación se describe el proceso:

1. Se identifica la clase mayoritaria del conjunto de entrenamiento junto a su volumetría.
2. Se repiten tantas veces como sea necesaria las listas de las clases minoritarias hasta alcanzar el tamaño de la mayoritaria. Luego, siguiendo la estrategias de generación de curvas sintéticas y/o perturbación de curvas reales, se diversifica cada curva de luz para que no sean exactamente iguales entre sí.
3. Para diversificar las clases minoritarias se utilizan las técnicas de aumento de datos descritas anteriormente, y se entrega al modelo las versiones modificadas de esta. De esta forma se asegura que el modelo vea distintas variaciones de las curvas, disminuyendo así la posibilidad de sobreajuste de las clases minoritarias.

A partir de la base de datos generada, el modelo es entrenado en base a lotes pequeños. Dado que el conjunto de datos está balanceado, en promedio, cada lote pequeño estaría balanceado

#### 3.3.2. Conjunto de validación

Para conservar las características de los datos originales, el conjunto de validación no es modificado siguiendo las técnicas de aumento de datos. Lo anterior se realiza para evitar sesgos asociados a dichas técnicas. Dado lo anterior, para balancear este conjunto cada curva de luz será ponderada según su clase. Este ponderador tiene la siguiente forma:

$$k = \frac{1}{c \cdot n}, \quad (3.1)$$

donde  $k$  es el ponderador,  $c$  es la cantidad de clases en estudio y  $n$  corresponde al número total de elementos de esa clase encontrados en el conjunto de validación. Este ponderador afectará en la función de pérdida.

### 3.3.3. Conjunto de prueba

Este conjunto es utilizado para determinar el desempeño de los modelos en base a los datos reales que se tengan. Dado lo anterior, no es posible balancear los tipos de supernovas a partir de aumento de datos. Es por esto, que para este conjunto no se realizará ninguna técnica para evitar el desbalance, sino que las evaluaciones correspondientes se realizarán de forma separada para cada tipo de supernova.

## 3.4. Imputación de datos

Por la naturaleza de como se obtienen las observaciones en astronomía, las series de tiempo (curvas de luz) suelen estar irregularmente muestreadas. Lo anterior trae consigo diversas dificultades como lo son: variabilidad en el número de observaciones por curva de luz, la existencia de largos periodos de tiempo sin observación que provocan la pérdida de información valiosa, entre otros.

Existen modelos astronómicos enfocados a la clasificación o modelación de curvas de luz que utilizan RNNs. Estos modelos manejan la irregularidad en el muestreo de los datos a través de la diferencia de tiempo existentes entre estos puntos [59]. A pesar de esto, las RNNs no son capaces de trabajar directamente con la irregularidad en los datos [27]. Lo anterior puede provocar que los resultados no generalicen bien el comportamiento de las diferencias de tiempo afectando así la tarea de predicción.

Con el fin de reducir la irregularidad en los datos, se plantea una metodología inspirada en [107] que realiza una imputación lineal entre observaciones conocidas. La idea de esto es lograr reducir el rango de las diferencias de tiempo para mejorar el modelo. Además, se escoge la imputación lineal debido a su simplicidad y que permite evitar técnicas que requieran conocer una cierta cantidad de puntos para que pueda funcionar. En otras palabras, la imputación lineal permite trabajar incluso con curvas de luz de solo dos puntos.

Para imputar los datos se siguen dos etapas. La primera etapa corresponde a discretizar el espacio. Para esto, se genera una grilla temporal de 1 día utilizando como punto de inicio la primera observación de la curva de luz. Una vez discretizado, se interpola linealmente entre observaciones conocidas. El objetivo de lo anterior es generar puntos en los días sin observaciones. Dado que las observaciones pueden ser afectadas en distintos momentos de la noche, la imputación se asegura de que existan variaciones dentro de la misma grilla. En otras palabras se asegura que exista una imputación por día, pero ésta puede ser a cualquier hora. Gracias a lo anterior, se reduce la irregularidad en el muestreo de los datos, donde se acota el rango de  $\Delta t$  a valores entre  $[1e^{-6}, 2 - 1e^{-6}]$  días, con un promedio y mediana de 1 día.

En la Fig. 3.9 se ilustra el procedimiento usado para generar una interpolación en una curva de luz.



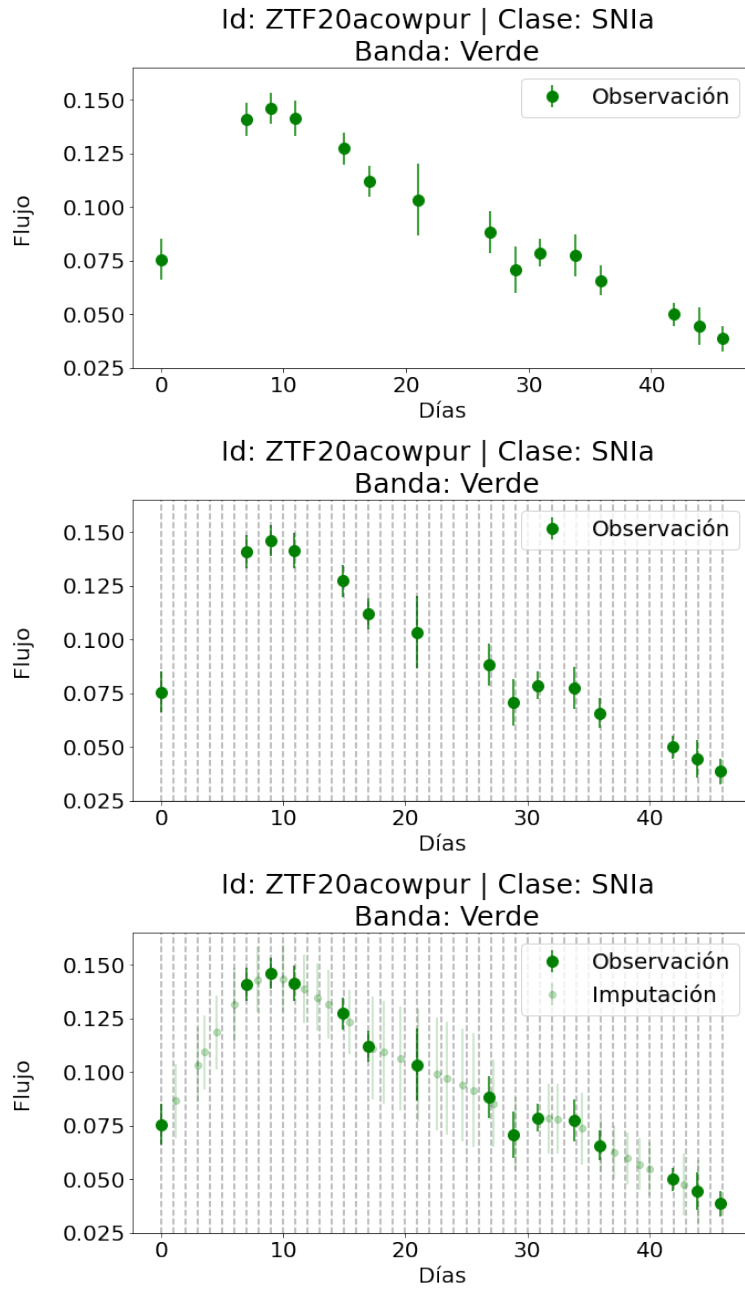


Figura 3.9: Proceso de imputación lineal de curvas de luz. (a) Muestra los datos observados, (b) muestra la discretización temporal centrada en 0 desde la primera observación en intervalos de 1 día, (c) muestra la imputación donde cada *bin* recibe la información de un punto.

### 3.5. Modelos Base

Con el objetivo de comparar las estrategias propuestas en este trabajo, se presentan los modelos bases a utilizar.

### 3.5.1. ALeRCE

El modelo ALeRCE corresponde a un predictor que se basa en el uso del modelo paramétrico de supernovas (SPM) propuesto en [14]. Para realizar las predicciones, se utiliza la función *curve fit* de la librería Scipy [108]. A partir de esta función (*curve fit*), el modelo SPM es ajustado en función de las observaciones por banda de cada curva de luz. Dado que este ajuste determina los parámetros del SPM, estos permiten estimar el comportamiento completo de la curva de luz. Lo anterior es utilizado para realizar predicciones. Se espera que mientras más información se tenga, mejores serán los resultados.

El proceso del ajuste de curva se resume de la siguiente forma:

- Definición de función del modelo: Esta corresponde al SPM.
- Definir restricciones de los parámetros: Estas se encuentran en [19].
- Estimar parámetros iniciales: Los parámetros iniciales se pueden estimar siguiendo los valores propuestos en [19].
- Ajustar datos: En base a las observaciones, función modelo, restricciones y adivinanza de los parámetros iniciales se realiza el ajuste de datos mediante la técnica de mínimos cuadráticos no lineal.

### 3.5.2. Naïve

En la tarea de predicción, es usual utilizar métodos ingenuos (*naive*) para comparar los resultados con los modelos propuestos [109]. Además, estos modelos son usados para independizarse de la escala de las métricas.

En este caso, el modelo ingenuo utilizado consiste en repetir el último valor observado para todo instante futuro. A partir de este modelo, se han escalado las métricas MAE [49] y las métricas RMSE (*root mean square error*), MIS (*mean interval score*) y PL (*pinball loss*) [102]. Además, en este trabajo se propone independizarse de la escala de la métrica CRPS (*continous ranked probability score*). En el mejor de nuestros esfuerzos no se encontró una versión escalada de CRPS en la literatura revisada.

En este trabajo de tesis, el modelo ingenuo solo se utiliza para independizar la escala de las métricas y no para compararla con las propuestas. Lo anterior se debe a que la simplicidad del modelo ingenuo no sería capaz de capturar las dinámicas de la curva de luz de supernovas. A pesar de lo anterior, gracias a la métrica MASE, el MAE del modelo ingenuo será comparado directamente con el MAE de los otros modelos.

## 3.6. Modelo

El modelo utilizado corresponde al propuesto por [74], donde se usa una red neuronal recurrente para parametrizar una función probabilística a elección. En este caso, se parametriza el promedio y desviación estándar de una distribución normal para cada instante de

tiempo asociado al horizonte de predicción. En términos matemáticos se busca obtener:

$$p(z_{i,t_0:T} | z_{i,1:t_0-1}, x_{i,1:T}), \quad (3.2)$$

donde  $[1 : t_0 - 1]$  es el rango de valores pasados (o rango de condición) y  $[t_0, T]$  es el rango de valores futuros (rango de predicción) de la serie de tiempo,  $z_{i,a:b}$  corresponde a las observaciones de la serie de tiempo de a hasta b y  $x_{i,1:T}$  corresponde a las covariables que se asumen conocidas en todo instante de tiempo.

El modelo, que desde ahora se denomina *Supernovae Forecaster* (SNeF) tiene una arquitectura del tipo *encoder - decoder*. Esta arquitectura permite a las RNNs mapear una secuencia de entrada hacia una secuencia de salida [74] En este caso, la RNN utilizada como *encoder* procesa la secuencia de datos de entrada y las proyecta en un vector de contexto. Este vector de contexto condiciona a la RNN utilizada como *decoder* para generar una secuencia de salidas deseadas.

Para la fase de entrenamiento, no se distingue entre *encoder - decoder*, dado que se conocen todos los valores asociados a los rangos de condición y predicción. Por otro lado, para la fase de predicción, si existe esta distinción: la codificación (*encoder*) se hace con respecto a los puntos conocidos, mientras que la decodificación (*decoder*) se hace para predecir usando el vector de contexto entregado por la codificación, las covariables y la predicción del punto anterior.

Lo anterior permite predecir de forma iterativa el comportamiento futuro de la serie de tiempo dado un cierto paso y horizonte de predicción. Cabe destacar, que ante iguales entradas, el modelo generará distintas predicciones. Esto se debe a que el modelo no predice un punto, sino que a partir de la distribución parametrizada, muestrea predicciones. Lo anterior, junto al muestreo de Monte Carlo, permite generar N trayectorias distintas ante la misma entrada, con las cuales se puede estimar la distribución de la predicción.

La Fig. 3.10 resume el esquema de entrenamiento y de predicción. Se destaca que la auto-regresión (la salida de un cierto instante es la entrada del siguiente instante) solo se realiza al predecir. Además, como se mencionó anteriormente, se muestra que la RNN parametriza una distribución normal, lo que permite generar diferentes trayectorias cada vez que se predice.

### 3.6.1. Entrenamiento

El entrenamiento sigue la estrategia de [74] cuya arquitectura se muestra en la Fig. 3.11. A continuación, se presentan los elementos más importantes del entrenamiento.

#### 3.6.1.1. Entradas

El modelo principal corresponde a una RNN que recibe como entrada características asociadas al instante de tiempo, junto a la predicción de la variable de interés: el flujo. En la etapa de entrenamiento, la variable de interés es conocida para todo instante de tiempo, por lo que éstas se usan directamente como entradas. Las características o covariables, deben ser conocidas para todo instante de tiempo (tanto para el entrenamiento como para la predicción). Dado lo anterior, a continuación se describen los grupos de covariables que pueden ser utilizados:

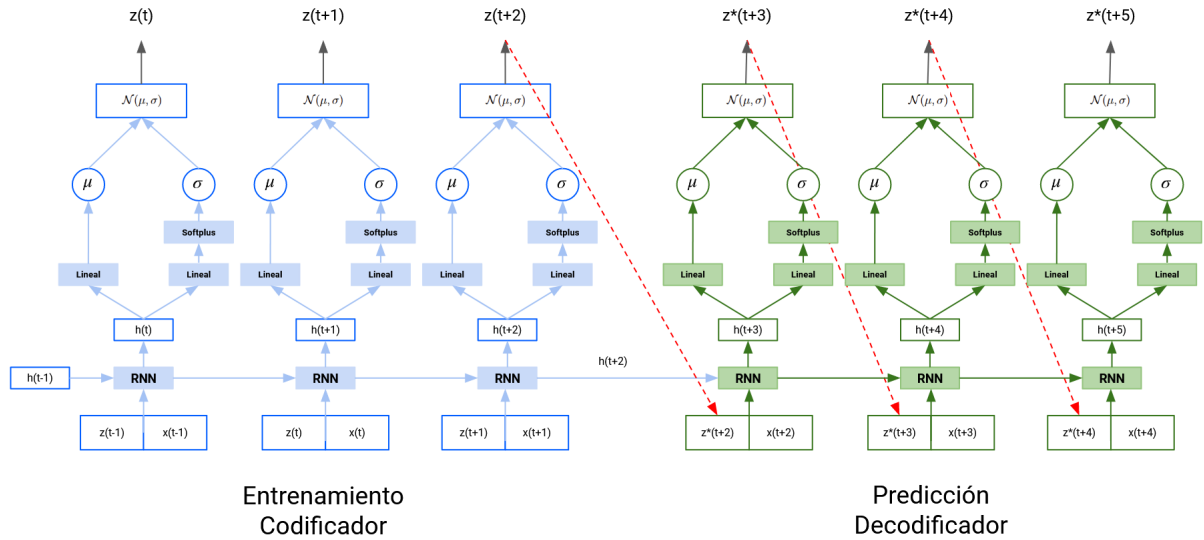


Figura 3.10: Esquema de entrenamiento y predicción, donde a la izquierda (en azul) se muestra la arquitectura del codificador y a la derecha (en verde) se muestra la arquitectura del decodificador. En ambos casos, se estima una distribución normal con la cual se muestrea para generar una predicción. Al repetir el proceso de predicción usando el muestreo de Monte Carlo, se obtiene la estimación de la distribución de las predicciones.

- **Covariables temporales:** Estas características son las que dependen del tiempo. Estas permitirían robustecer y mejorar el desempeño de la predicción. Un ejemplo de esta característica es la diferencia de tiempo existente entre observaciones.
- **Covariables atemporales:** Estas características son utilizadas para que el modelo pueda acentuar las relaciones entre las curvas del mismo tipo. La información sobre la clase y la banda son ejemplos de este tipo de covariables.

Para que las características tengan un mismo nivel de importancia en el modelo, se aplica una normalización a los datos que lo requieran. Esta normalización tiene la siguiente forma:

$$c_{norm}(t) = \frac{c(t) - \mu_t}{\sigma_t}, \quad (3.3)$$

donde el  $c_{norm}(t)$  corresponde a la característica normalizada,  $c(t)$  corresponde a la característica,  $\mu$  y  $\sigma$  corresponden al promedio y desviación estándar de todos los valores asociados a dicha característica en el conjunto de entrenamiento.

### 3.6.1.2. Valor deseado

Para entrenar un modelo de predicción de series de tiempo, se requiere conocer el valor deseado que se quiere predecir. En general, se busca predecir el valor futuro de la serie de tiempo asociado al instante  $t + 1$  dado la información del instante  $t$ . Dado lo anterior, se suele utilizar la observación de la misma serie de tiempo del instante  $t + 1$  como valor deseado para la observación en  $t$  en el entrenamiento.

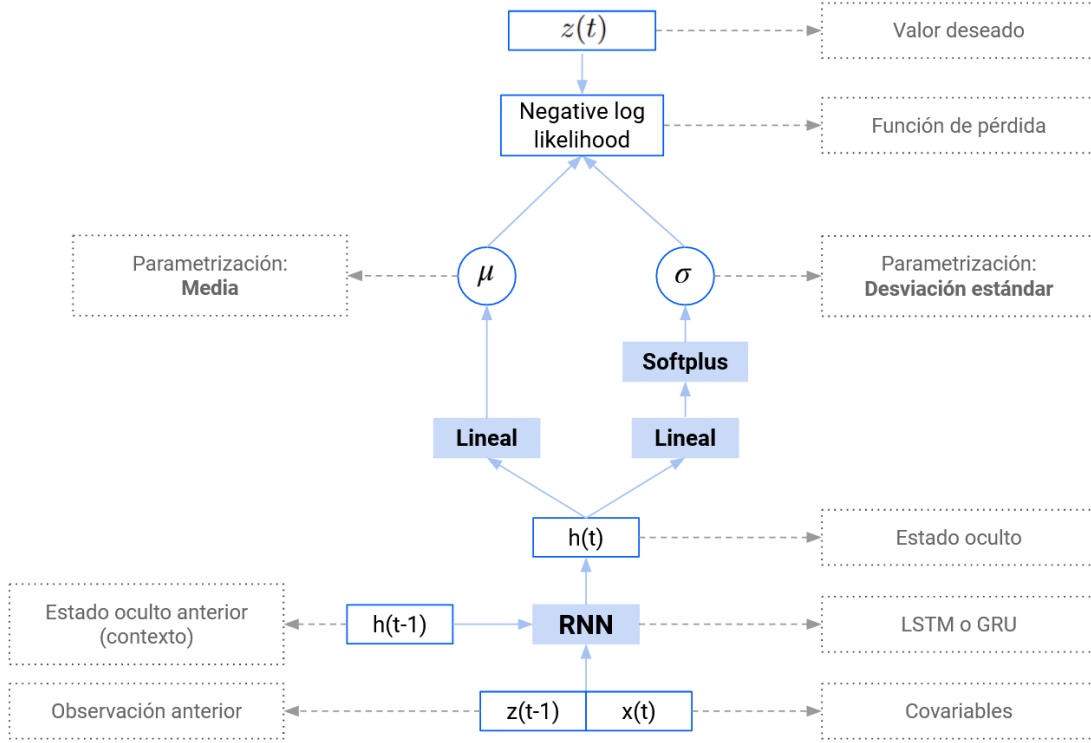


Figura 3.11: Arquitectura del codificador junto al esquema de entrenamiento. La arquitectura utilizada consta de una RNN que obtiene los valores  $\mu$  y  $\sigma$  de una distribución normal. Para el caso de  $\mu$  se obtiene a partir de una capa lineal, mientras que para  $\sigma$  se le agrega una capa *softplus* para asegurar que este valor siempre sea positivo. Estos valores entran a una función de pérdida denominada *negative log likelihood* que se compara con el valor deseado  $z(t)$

Para el caso de las curvas de luz, las observaciones poseen un error de medición. Este error también afecta al valor deseado, lo que podría ocasionar que el modelo no aprenda correctamente los patrones de las muestras. Dado lo anterior, se propone estimar los valores deseados a partir del ajuste de los datos de observación al modelo paramétrico de supernovas utilizando las técnicas *curve fit* (Ver Sección 3.5.1 para más detalles) y MCMC.

Para el caso de las curvas generadas de forma sintética, se utiliza como valor deseado, las curvas ajustadas por el modelo MCMC. Mientras que para el caso de los datos reales, se utilizan las curvas ajustadas por el modelo *curve fit*. A partir de lo anterior, se espera que el modelo se comporte como un *denoising autoencoder*, es decir, una arquitectura que recibe entradas ruidosas y arroja salidas limpias.

### 3.6.1.3. Función de pérdida

Para entrenar al modelo de predicción, se debe definir una función de pérdida que compare las predicciones realizadas por el modelo con los valores esperados. Para esto, se utiliza la función *negative log likelihood gaussiana*, modificada en base a [19, 59]. A continuación se presenta la ecuación y posterior a eso se describen las modificaciones:

$$Loss(\mu, \sigma, y) = -\frac{1}{N_{lc}} \sum_{lc=0}^{N_{lc}} \left( \frac{1}{2} \sum_{t=0}^{N_{lc,t}} \frac{1}{(1+\alpha\theta_{lc,t}^2)} \left( \ln(2\pi) + \ln(\sigma_{lc,t}^2) + \frac{(y_{lc,t} - \mu_{lc,t})^2}{\sigma_{lc,t}^2} \right) \right), \quad (3.4)$$

donde  $lc$  es la curva de luz,  $N_{lc}$  es el número de curvas de luz por *mini batches*,  $N_{lc,t}$  es la cantidad de puntos de la curva de luz,  $y_{lc,t}$  es el valor deseado,  $\sigma_{lc,t}^2$  y  $u_{lc,t}$  corresponden a la parametrización de una distribución normal, siendo  $\sigma$  la desviación estándar y  $\mu$  el promedio. Es importante notar que estos resultados se obtienen considerando solo los puntos reales, es decir, ignorando los puntos imputados. Además, dentro de la función de pérdida se agrega un factor penalizador siguiendo  $\frac{1}{1+\alpha\theta_{lc,t}^2}$  [19,59] donde  $\alpha$  es un hiperparámetro ajustable fijado en  $1e-3$  y  $\theta_{lc,t}$  es el error asociado al error de la observación. Lo anterior se realiza para que el error de observación sea considerado sin afectar considerablemente a la función de pérdida.

### 3.6.1.4. Optimización

Una estrategia para entrenar un modelo es pre entrenarlo y luego realizar un ajuste fino [19,41]. La primera etapa (pre-entrenamiento) busca armar un modelo basado en datos que no necesariamente estén etiquetados. En esta etapa el modelo busca determinar patrones generales con los que puede completar una tarea. La segunda etapa (ajuste fino) busca aterrizar el conocimiento a una tarea en particular con características particulares. A continuación se describe cada una de las etapas junto al optimizador utilizado.

Para el pre-entrenamiento, se utilizan las curvas sintéticas generadas a partir de la técnica MCMC (Ver Sección 3.2.1). En esta etapa, el modelo recibe una gran diversidad de muestras basadas en el ajuste de los datos al modelo paramétrico de supernovassiguiendo a la técnica de MCMC. Este procedimiento es llevado a cabo a través del optimizador Adam. Este optimizador posee los siguientes hiperparámetros:  $betas = (0,9, 0,999)$ ,  $weightdecay = 5e-4$  y una tasa de aprendizaje ( $lr$ ) que varía linealmente, a través de un *learning rate warm-up schedule*, desde un  $lr_{min} = 1e-10$  a  $lr_{max} = 2e-4$ . Además, se utiliza un  $batchsize = 128$  y se restringe la norma global del gradiente a 1 según la sugerencia dada en [110].

Para el ajuste fino, se utilizan las curvas generadas en base al aumento real (Ver Sección 3.2.2). En este caso, el modelo es ajustado con curvas perturbadas en base a las observaciones reales. Este procedimiento, al igual que el anterior, es llevado a cabo a través del optimizador Adam. Este optimizador utiliza hiperparámetros similares al del pre-entrenamiento:  $betas = (0,9, 0,999)$  pero con una tasa de aprendizaje menor:  $lr = 1e-4$ . Además, a diferencia de la fase de pre-entrenamiento, el tamaño del mini lote es inferior:  $batchsize = 32$ . Lo anterior se realiza siguiendo la idea de que a menor tamaño del mini lote, los resultados alcanzados presentan un mejor desempeño [111].

## 3.6.2. Predicción

Como fue mencionado anteriormente, esta estrategia de predicción se basa en [74]. En la Fig. 3.12 se observa el detalle de la arquitectura utilizada. A continuación se detallan los elementos más importantes de la predicción.

### 3.6.2.1. Predicciones probabilísticas

El procedimiento para realizar las predicciones se describe a continuación:

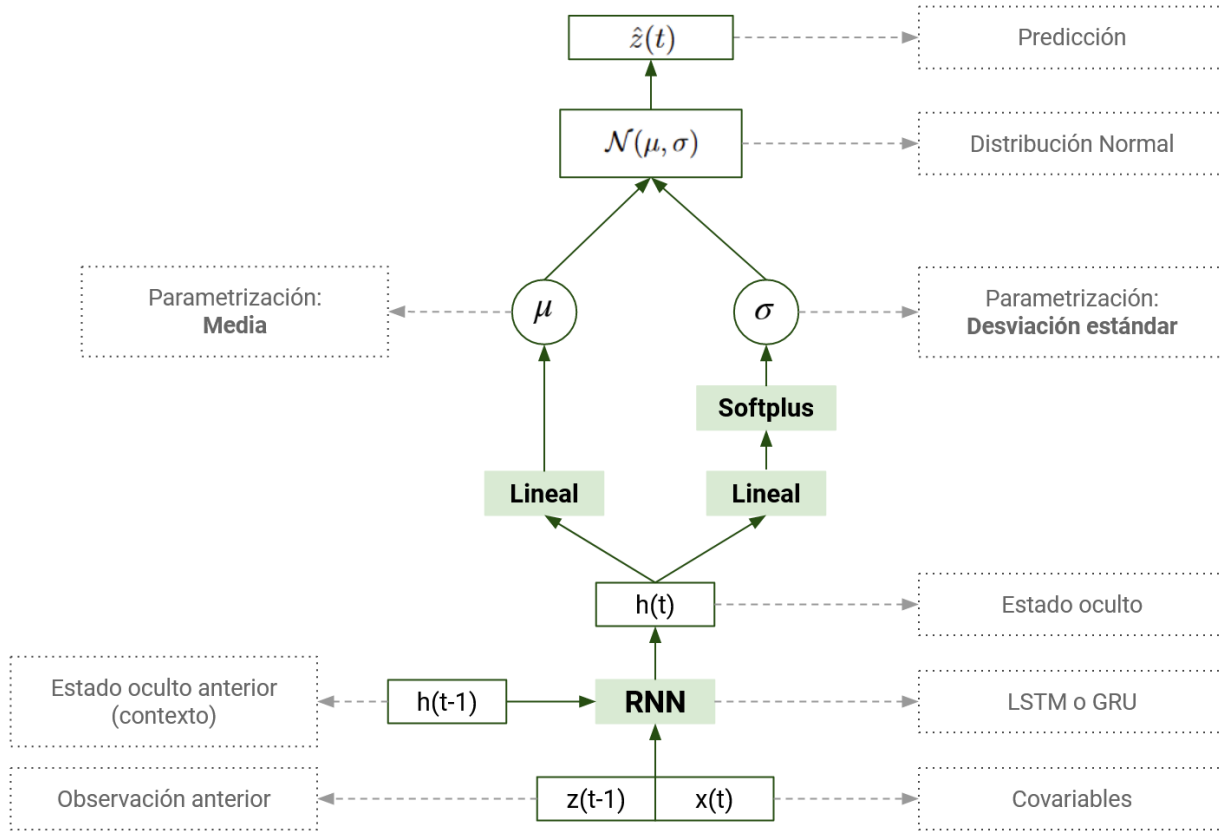


Figura 3.12: Arquitectura del decodificador junto al esquema de predicción. La arquitectura utilizada consta de una RNN que obtiene los valores  $\mu$  y  $\sigma$  de una distribución normal. Para el caso de  $\mu$  se obtiene a partir de una capa lineal, mientras que para  $\sigma$  se le agrega una capa *softplus* para asegurar que este valor siempre sea positivo. Estos valores permiten muestrear las predicciones de forma iterativa, a partir de una distribución normal  $N(\mu, \sigma)$ . Estas predicciones ( $\hat{z}(t)$ ) variarán por cada ejecución de la predicción realizada.

1. La información de las observaciones y covariables conocidas será compactada en un vector de contexto obtenido en base al codificador.
2. El vector de contexto junto a las covariables conocidas futuras serán utilizadas como entrada para el decodificador.
3. En base a los valores entregados, el modelo determinará los valores  $\mu$  y  $\sigma$  de una distribución normal, que posteriormente será utilizada para muestrear una predicción.
4. Utilizando un método iterativo, cada predicción será utilizada como entrada para predecir el siguiente instante de tiempo deseado.

Cabe notar, que al repetir este proceso, las predicciones no serán exactamente las mismas debido a que éstas son generadas siguiendo una distribución normal. A partir de lo anterior, es posible repetir este procedimiento  $N$  veces obteniendo así  $N$  trayectorias distintas de predicción. Lo anterior permitiría estimar la distribución de probabilidad de las predicciones, generando así una predicción probabilística. El procedimiento descrito anteriormente es más conocido como técnica de muestreo de Monte Carlo.

### 3.6.2.2. Propuesta para considerar error de observación

Como ya se ha mencionado anteriormente, las observaciones astronómicas poseen un error de observación asociado. Para considerar este error, en cada iteración de predicción se perturbarán los flujos conocidos que entran al codificador. Lo anterior provocará que se generen vectores de contextos diferentes entre sí. Esto permitirá que la predicción probabilística considere la información de la incerteza de los datos entrantes.

La perturbación de los flujos se realiza asumiendo que las mediciones siguen una distribución normal. En otras palabras, la observación es considerada como el promedio de la distribución y el error de observación como la desviación estándar.

A partir del procedimiento anterior se espera que los intervalos de predicción aumenten su tamaño. Esto permitiría capturar de forma más robusta los límites superiores e inferiores de las predicciones.

## 3.7. Evaluación

Evaluar el desempeño de los modelos de predicción sirve para comparar el desempeño de estos. La evaluación es una tarea sencilla en la que se busca comparar las predicciones con los valores esperados a partir de las métricas. A pesar de lo dicho anteriormente, en este trabajo se tiene una serie de dificultades relacionadas a los datos de las curvas de luz que complejizan la tarea de evaluación. Las principales dificultades son listadas a continuación:

- Largos *gaps* de tiempo: Existen períodos en los que no hay observaciones. Esto dificulta la tarea de evaluar dado un cierto horizonte de predicción.
- No están alineadas: La primera observación de la curva de luz puede darse antes, durante o después del momento de máximo brillo. Esto dificulta la tarea de comparar los resultados entre distintas curvas de luz.
- Escala entre curvas de luz: Las curvas de luz están en distintas escalas. Esto dificulta la tarea de comparar los resultados entre distintas curvas de luz.
- Escala interna: La curva de luz de las supernovas se caracteriza por poseer grandes cambios en su brillo. Lo anterior se traduce en que dependiendo del momento de brillo que se esté evaluando, se tendrán errores con diferentes escalas.

Para enfrentar estas dificultades, se propone utilizar el SPM para obtener una aproximación continua de la curva de luz. A partir de lo anterior, se obtendrán los valores deseados para cualquier instante de tiempo, permitiendo así comparar las predicciones con estos valores. Se debe tener en cuenta, que este ajuste es confiable solo entre la primera y última observación entregada. Para considerar esto, solo se evaluará sobre curvas de luz cuyo horizonte de predicción estén entre la primera y última observación conocida.

Por otro lado, para enfrentar el problema de las escala de los flujos, se utilizan métricas independientes de escala mostradas en la Sección 2.8.

Es importante indicar que para evaluar se debe considerar tanto el paso como el horizonte de predicción. La evaluación se hará sobre un horizonte de 7 días. Esto se realiza dado



que en general la cadencia de observación suele ser menor, por lo tanto, las predicciones se actualizarían antes de estos 7 días. Además, esto se realiza con el fin de que la comparación del modelo se enfoque solo en la primera semana.

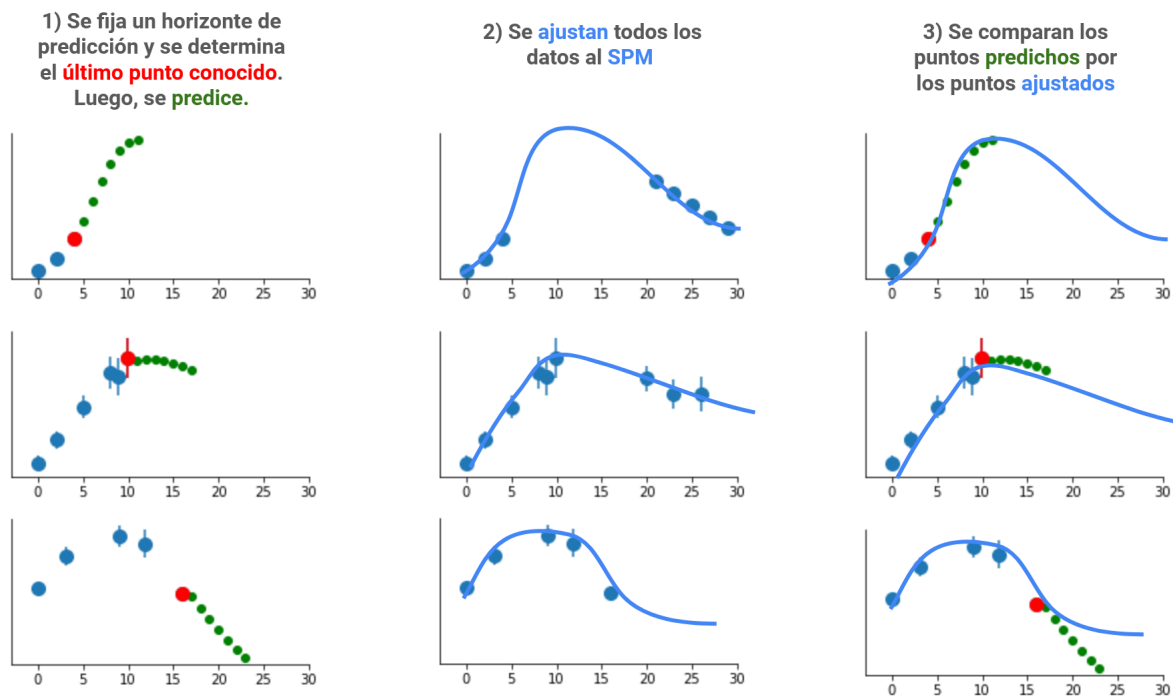


Figura 3.13: Esquema de evaluación final. Los gráficos de la izquierda muestran los valores predichos en un horizonte de predicción de 7 días, a partir del último punto conocido. Los gráficos centrales muestran el proceso de ajuste de valores deseados con los que será comparada la predicción. Los gráficos de la derecha muestran la comparación realizada entre la predicción y los valores ajustados.

La Fig. 3.13 muestran los pasos para comparar entre las predicciones con los puntos ajustados por el modelo SPM. Para evaluar se fija un horizonte de predicción y se determina el último punto conocido a partir del cual se generan las predicciones. Paralelamente, se realiza un ajuste de los datos al modelo SPM. Es importante notar que este ajuste solo se realiza para comparar, por lo que el predictor no se ve influenciado por esta etapa. Finalmente, se comparan los puntos predichos con los puntos ajustados.

A continuación se describen dos propuestas de evaluación. La primera estudia el comportamiento de las predicciones considerando solo la cantidad de observaciones entregadas como entrada. La segunda evaluará el desempeño según la cantidad de observaciones existentes en un cierto periodo de tiempo.

### 3.7.1. Estrategia número de puntos

Esta estrategia consiste en evaluar el desempeño del modelo utilizando una misma cantidad de puntos entrantes de las curvas de luz, sin considerar la diferencia de tiempo entre estos. De esta forma se evalúa el desempeño del modelo según su capacidad para predecir en base a la cantidad de observaciones.

Por ejemplo, si se evalúa en base a las primeras 5 observaciones, el modelo comparará las predicciones que ocurran después del quinto punto de cada supernova con el paso y horizonte de predicción anteriormente mencionado.

Dado que no es lo mismo predecir con 5 puntos que están distribuidos a lo largo de 10 días que evaluar con puntos que están distribuido a lo largo de 30 días, es que se propone una segunda estrategia de evaluación. Esta se describe a continuación.

### 3.7.2. Estrategia umbral de días

Debido a los problemas mencionados anteriormente, se dificulta comparar el comportamiento del modelo en base a las distintas curvas de luz. Dado lo anterior, se propone una estrategia que se basa en aplicar umbrales de días. Esta técnica de evaluación, busca analizar el comportamiento de las curvas de luz que poseen observaciones hasta este umbral. Los pasos que se siguen en esta estrategia están explicados a continuación y ejemplificados en la Fig. 3.14.

- Se define y aplica un umbral de días. Este es aplicado a cada curva de luz del conjunto de prueba.
- Se identifica el último punto conocido que esté dentro del umbral propuesto.
- Se define un horizonte y paso de predicción. Siguiendo los valores anteriores, se genera las predicciones al paso dado hasta el horizonte de predicción dado.

### 3.7.3. Métricas

Las métricas con las que se determina el rendimiento de las estrategias en estudio corresponden a las mostradas en Sección 2.8. Como ya se mencionó, se debe realizar una comparación entre las predicciones puntuales y probabilísticas.

Dado que el método propuesto entrega predicciones probabilísticas se debe establecer una forma para encontrar una predicción puntual. Es por esto, que se trabajará con la mediana de la distribución de predicciones por cada instante de tiempo a predecir. En resumen, se comparará el valor deseado con la mediana de la distribución de los datos.

Las métricas anteriores son escaladas (excepto la métrica PICP) en base a un modelo *naive*. Este modelo asume que el futuro de la curva es igual al último valor observado. La ventaja de realizar esto, es que las métricas son independientes de las escalas y, por ende, son más fáciles de comparar.

### 3.7.4. Evaluación para 5 conjuntos de datos diferentes

Con el fin de asegurar que el modelo propuesto posee un desempeño similar al cambios los conjuntos de entrenamiento, evaluación y prueba, se realiza una validación cruzada. Para esto, el conjunto de datos completo se separa en 5 grandes lotes de datos. Lo anterior significa que el modelo es entrenado, evaluado y probado en base a 5 conjuntos de datos diferentes.

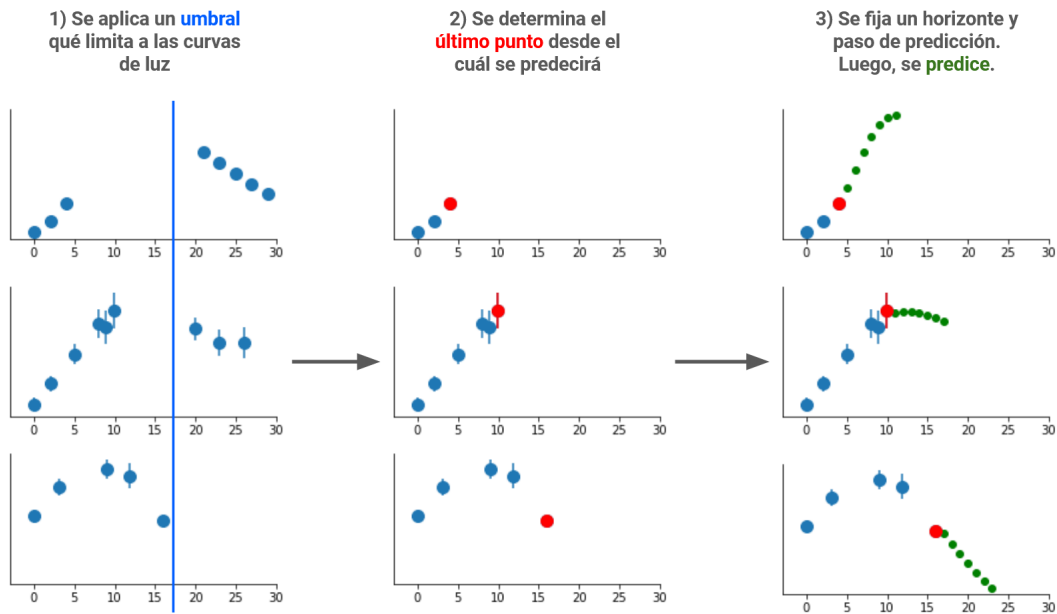


Figura 3.14: Ejemplos de estrategia umbral: En la figura se observan 3 curvas de luz que pasan por los 3 pasos de la estrategia de umbral de días. En el primer paso se aplica un umbral, el cuál se representa por la línea vertical azul. Posterior a eso, se determina el último punto (señalado con rojo) para cada una de las curvas de luz, donde se observa que no coinciden entre sí según el eje x. Finalmente, a partir del último punto se genera la predicción (puntos verdes) a un paso de 1 día con un horizonte de predicción de 7 días.

Además, para que los resultados sean más representativos y no tengan tantos sesgos, se realiza la partición de los datos de tal forma que los datos de los 5 conjuntos de validación y prueba sean diferentes entre sí.

## 4. Resultados, análisis y discusión

En este capítulo se presentan los resultados experimentales junto a su análisis y discusión. Las estrategias de evaluación corresponden a las presentadas en la sección 3.7. Los experimentos que se presentan a continuación son evaluados utilizando un horizonte de predicción de 7 días a un paso de 1 día. Además, como se explicó anteriormente, para obtener la distribución de la predicción se utiliza un muestreo de Monte Carlo donde se repite el proceso de predicción durante  $N$  veces. Se utiliza un  $N$  de 2000 iteraciones.

### 4.1. Comparación de modelos

En esta sección se muestra la comparación de las métricas resultantes del modelo base y las diferentes versiones del modelo propuesto. Para evaluar, se utilizan las técnicas propuestas en la Sección 3.7.

A continuación se presenta una breve descripción del modelo ALerCE (ALerCE - P) y las diferentes versiones del modelo propuesto (SNeF) que serán comparadas:

- **ALerCE - P:** Modelo utilizado como *baseline*. Este modelo realiza predicciones puntuales, por lo que solo se comparará con las métricas MASE y sMAPE.
- **SNeF - SNIa:** Modelo entrenado solo con curvas de luz de supernovas de tipo Ia. Este modelo fue entrenado con la función de pérdida afectada por el error, con datos imputados linealmente y su valor deseado correspondiente al ajustado con SPM.
- **SNeF - Entrenamiento:** Modelo entrenado con curvas de luz de supernovas de todo tipo (Ia, Ibc, II y SLSN). Este modelo fue entrenado exactamente igual que el modelo anterior (SNeF - SNIa).
- **SNeF - Predicción:** Este modelo tiene las mismas especificaciones que el modelo SNeF - Entrenamiento, solo que se agrega la propuesta de perturbar los valores entrantes (Ver Sección 3.6.2.2).
- **SNeF - Covariables:** Este modelo tiene las mismas especificaciones que el modelo SNeF - Predicción, pero en su entrenamiento se agrega el flujo de la primera observación y el logaritmo de la distancia temporal con respecto a la primera observación. Esta idea surge, dado que en [19] se mostró que para la tarea de clasificación, la primera observación posee una gran relevancia al clasificar. Se espera que para la tarea de predicción esta covariable mejore el desempeño del modelo.

Como es posible notar, cada uno de estos modelos corresponde a una versión modificada y/o mejorada de su antecesor (siguiendo el orden de la lista). A partir de lo anterior, es posible entender cómo afecta cada una de las propuestas de entrenamiento, de predicción y agregación de covariables al desempeño del modelo.

#### 4.1.1. Especificaciones del entrenamiento

Como se mencionó anteriormente, cada uno de los modelos corresponden a una versión mejorada del anterior. En otras palabras, todos los modelos fueron entrenados siguiendo las mismas lógicas. A continuación se muestra una lista de cada uno de los elementos que se consideran al entrenar los modelos:

- Pre entrenamiento con datos sintéticos: Se genera un total de 16 supernovas sintéticas (Ver Sección 3.2.1) por cada una existente en el conjunto de entrenamiento.
- Ajuste fino con datos reales aumentados: Se genera un total de 16 supernovas en base a perturbaciones (Ver Sección 3.2.2) por cada una existente en el conjunto de entrenamiento.
- Imputación lineal: Se realiza una discretización y posterior imputación lineal de datos (Ver Sección 3.4) con el objetivo de que en cada día se tenga una observación. Es importante destacar que esta imputación solo se realiza entre observaciones y es utilizado para que la RNN no pierda continuidad temporal.
- Hiperparámetros: Se utilizan los mismos para todos los casos. Estos fueron detallados en la Sección 3.6.1.4.
- Celda LSTM: Los principales experimentos se realizan con la celda LSTM. En la Sección B se muestra una comparación de los resultados entre la celda LSTM y GRU.
- Valor deseado: Se utiliza como valor deseado el valor estimado por el modelo paramétrico mediante las técnicas de MCMC o *curve fit*. Para el caso de los datos sintéticos, los valores deseados corresponden a los obtenidos por el MCMC, mientras que para el caso de los datos reales se utilizan la técnica de *curve fit*.
- Número de épocas: El número de épocas se fijó en 100 tanto para el pre entrenamiento como para el ajuste fino. Se puede detener el entrenamiento si a lo largo de 25 épocas no mejora la función de pérdida del conjunto de validación.

#### 4.1.2. Supernovas de tipo Ia

En esta sección se evalúa el desempeño del modelo sobre las curvas de luz de las supernovas de tipo Ia. La evaluación se realiza en base a 6 métricas obtenidas para las dos estrategias de evaluación propuestas. A continuación se presenta un análisis y discusión para cada una de métricas.

#### 4.1.2.1. MASE

En las Figuras 4.1a y 4.2a se observa que todas las versiones del modelo propuesto (SNeF) tienen un MASE inferior a 1. Lo anterior significa que el desempeño de cada versión es superior al modelo *naive*. Además, se observa que todas las variaciones del modelo SNeF presentan un desempeño similar entre sí y superan significativamente al del modelo ALeRCE-P.

La Figura 4.1a muestra que los modelos tienden a mejorar a medida que el número de observaciones aumentan. A pesar de lo anterior, al medir el desempeño del modelo cuando se tienen cerca de 25 observaciones, se observaría un leve empeoramiento. Este empeoramiento podría estar gatillado por la disminución de muestras válidas para evaluar a medida que se tiene un mayor número de puntos. Otra razón podría ser que, al tener una gran cantidad de observaciones, un gran número de supernovas alcanzaría su fase de extinción. Esta fase de extinción implicaría que los últimos puntos podrían ser muy similares entre sí, provocando que la métrica MASE alcance valores muy altos.

En la Figura 4.2a se muestra que las variaciones del modelo SNeF presentan un comportamiento particular. Se destaca la existencia de un empeoramiento del desempeño del modelo al evaluar según un umbral de 11 días. Este empeoramiento se debería a que la mediana de las muestras de luz de las SNIa alcanzan su máximo brillo después de 12 días de su primera observación. Lo anterior indicaría que el desempeño del modelo se vería afectado según si se tiene como entrada el momento de su máximo brillo.

#### 4.1.2.2. sMAPE

El sMAPE es una métrica que permite medir el error relativo de cada una de las predicciones, lo que evitaría problemas de escalas asociados a la misma curva de luz.

En las Figuras 4.1d y 4.2d se observa que todas las versiones del modelo propuesto (SNeF) presentan un desempeño similar entre sí y superan al modelo ALeRCE-P. Además, en este caso se observa que en ambas figuras luego de superar los primeras observaciones o los umbrales de días, el desempeño del modelo suele ser similar para todos los casos.

En la Fig. 4.2d se observa un estrechamiento de la diferencia del sMAPE entre ALeRCE-P y las variaciones propuestas. Lo anterior podría deberse a que al utilizar un umbral de días cercanos a 100, la mayoría de las supernovas se encontrarían en su momento de extinción, provocando que ambos modelos (ALeRCE-P y propuesto) tiendan a tener un comportamiento similar.

#### 4.1.2.3. PICP y MSIS

Dado que según las métricas MASE y sMAPE no se observaron diferencias significativas entre los diferentes modelos de observación, se complementa el estudio en base a las métricas PICP y MSIS, las cuales miden la confianza y tamaño del intervalo de un cierto intervalo de predicción. En este caso se obtienen los resultados referentes al intervalo de predicción del 95 % dado que este capturaría la mayor parte del intervalo, sin considerar casos bordes.

Las Figuras 4.1b y 4.2b presentan el desempeño del modelo según su PICP. Esta métrica indicaría la confianza de las predicciones del modelo, el cual podría entenderse como el

porcentaje de observaciones reales que son capturadas por el intervalo de predicción del modelo.

Por otro lado, en las Figuras 4.1e y 4.2e se presenta la métrica MSIS. Esta métrica indicaría el tamaño del intervalo de predicción, la que además penaliza según si la observación se encuentra dentro de un intervalo de predicción dado.

Al comparar los resultados según ambas estrategias de evaluación, no se observan diferencias referentes al desempeño relativo de cada una de las variaciones del SNeF.

Los resultados asociados al modelo SNeF - SNIa presentan las métricas de desempeño PICP y MSIS más bajas de todas las versiones del modelo. Lo anterior significa que este modelo privilegia poseer intervalos de predicción más pequeños a un costo de dejar un mayor número de elementos fuera. Al evaluar según el número de puntos la confianza (evaluada por el PICP) presentaría un promedio estable, pero una varianza que aumenta a medida que el número de observaciones crece. Al evaluar según el umbral de días, a medida que se acerca a un umbral de 100 días, el desempeño tiende a empeorar.

El modelo que le sigue en cuanto a desempeño corresponde al SNeF - Entrenamiento, el cual alcanza valores de PICP superiores que el modelo SNeF - SNIa, y mantiene un intervalo de predicción de un tamaño similar que el caso anterior. Esto significa, que entrenar el modelo con todos los tipos de supernovas, permite al modelo capturar de mejor forma todas las posibles formas que podría tener la predicción.

Luego, se observa que los modelos SNeF - Predicción y SNeF - Covariables poseen un mejor desempeño en cuanto a PICP (alcanzando valores superiores al 97,5%, a costo de poseer un intervalo de predicción más alto).

Al comparar el modelo SNeF - Entrenamiento con el SNeF - Predicción se tiene que la única diferencia radica en que en el segundo se agrega una perturbación asociada al error de observación a la entrada del modelo. Esta perturbación permitiría mejorar la confianza del modelo al agrandar el intervalo de predicción.

Luego, al comparar el modelo SNeF - Predicción con el SNeF - Covariables se tiene que la única diferencia se da que el segundo se entrena con dos covariables nuevas asociadas a la primera observación. Esta covariable permitiría mejorar la confianza del modelo según las primeras observaciones y reduciría levemente el tamaño del intervalo de predicción.

#### **4.1.2.4. CRPSS y SPL**

Las Figuras 4.1c, 4.1f, 4.2c y 4.2f resultan poseer un comportamiento bastante similar entre sí. Estas métricas indicarían que los modelos SNeF - SNIa y SNeF - Entrenamiento tienen un comportamiento bastante similar entre sí. Lo anterior significa, que en cuanto a estas métricas, entrenar con un tipo de curva de luz o con distintos tipos no varía los resultados de forma significativa.

Luego, comparando el modelo SNeF - Covariables y SNeF - Predicción se observa que tienen un peor desempeño que los modelos anteriores. Además, para los primeras observaciones ambos modelos tiende a funcionar similarmente. A medida que se tienen más observaciones, el modelo SNeF - Covariables presenta un mejor desempeño. Lo anterior indica que agregar la covariable asociada a la primera observación mejora el desempeño del modelo.

### 4.1.3. Resumen de las métricas para SNIa

En base al análisis de las métricas anteriores, es posible indicar que el modelo propuesto tiene un desempeño superior al modelo ALerCE-P en cuanto a sus métricas puntuales. Además, se observa que las distintas variaciones del modelo SNeF tienen un desempeño puntual similar.

Las diferencias de las versiones del modelo SNeF se centran en la confianza y tamaño del intervalo de predicción en uso. Se observa que los modelos más confiables son los que poseen un mayor intervalo de predicción. En este caso, los modelos más confiables corresponden al SNeF - Predicción y SNeF - Covariables cuyo desempeño tiende a ser similar. A pesar de lo anterior, y según las métricas probabilísticas CRPSS y SPL, el modelo SNeF - Covariables presenta un mejor desempeño probabilístico.

Al comparar el desempeño según el tipo de celda recurrente (Ver Anexo B), se observa que la red LSTM tiende a funcionar levemente mejor que la red GRU.

### 4.1.4. Resumen de las métricas para las clases SNII, SNIbc y SLSN

Con el objetivo de no ser redundante en el análisis, las métricas de desempeño del modelo sobre las SNII, SNIbc y SLSN son mostradas en Anexo A. Además, se compara el comportamiento del modelo ALerCE-P, el modelo *SNeF* - Tipo (modelo entrenado solo con las supernovas de un tipo, pero considerando el flujo de la primera observación junto a la distancia entre las distintas observaciones con respecto a la primera) y el modelo *SNeF* - Covariables. A continuación se presenta un resumen de las métricas.

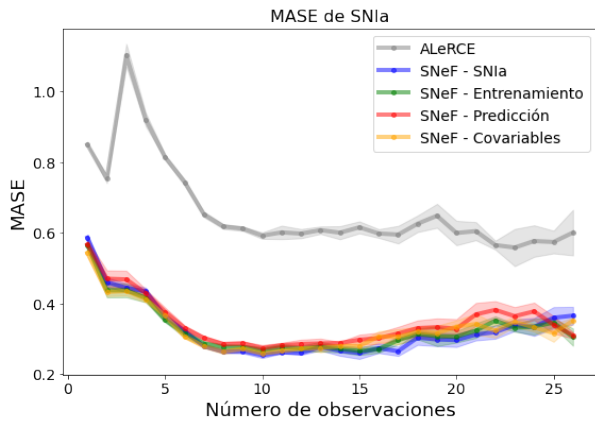
Para el caso de las SNII, las variaciones del modelo SNeF presentan un mejor desempeño puntual que el modelo ALerCE-P. Además, dado que su MASE es inferior a 1, significa que el modelo propuesto es mejor que el modelo *naive*. Por otro lado, según la métrica PICP, ambos modelos tendrían una confianza similar, pero el modelo entrenado solo con un tipo de supernovas tendría el menor tamaño de su intervalo de predicción. Lo anterior indica que solo entrenando con un tipo de supernova se tendría un leve mejor desempeño. Finalmente, a partir de las métricas probabilísticas CRPSS y SPL se tiene que el modelo SNeF - SNII presenta un mejor comportamiento que el SNeF - Covariables.

Para el caso de las SNIbc y SLSN el comportamiento de las métricas tiende a ser más ruidoso dado la escasa cantidad de muestras utilizadas para medir el desempeño de los modelos. A pesar de lo anterior, se observa que en general el modelo entrenado solo con un tipo de estas supernovas tiende a tener un mejor comportamiento que el entrenando con todas. Lo anterior indica, que a pesar de que se utilizaron técnicas de balanceo de clases, éstas no permitieron capturar completamente la dinámica de los modelos de clases minoritarias.

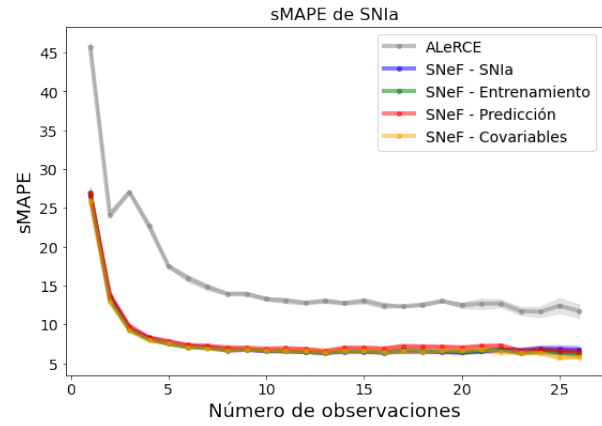
Al comparar los desempeños puntuales de los modelos propuestos con ALerCE se obtiene que en la mayoría de los casos de las SNIbc, los modelos propuestos tiende a tener un mejor desempeño que el modelo ALerCE-P. Mientras que para el caso de las SLSN, solo para cuando se tienen pocas observaciones el modelo propuesto tiende a tener un mejor comportamiento.

Al comparar el desempeño según el tipo de celda recurrente (Ver Apéndice B), se observa que la red LSTM tiende a funcionar levemente mejor que la red GRU.

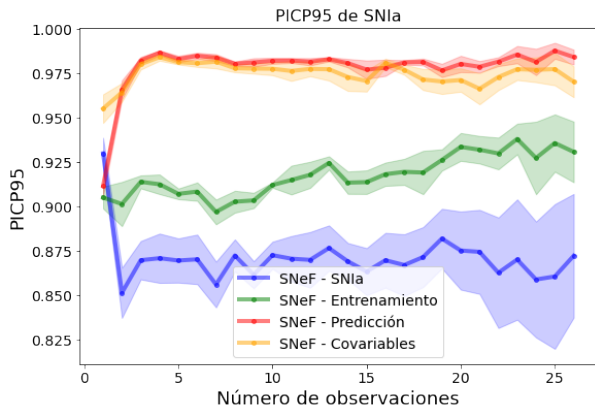




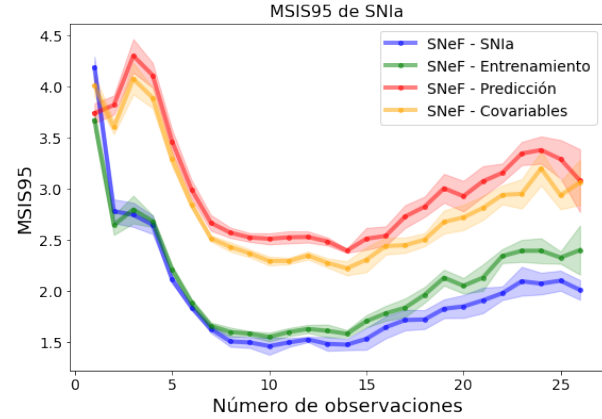
(a) MASE



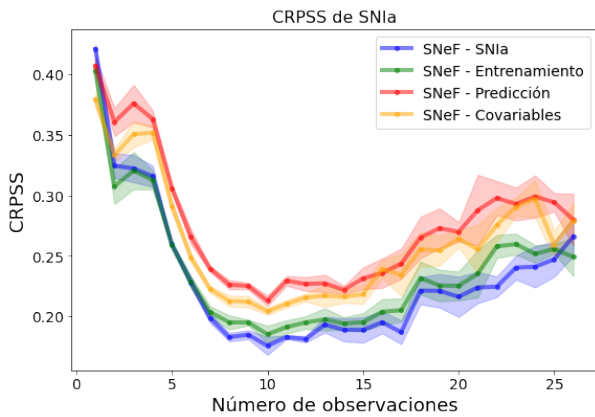
(d) sMAPE



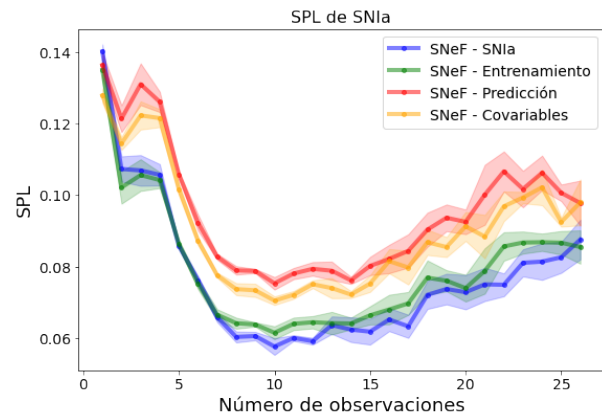
(b) PICP



(e) MSIS



(c) CRPSS



(f) SPL

Figura 4.1: Conjunto de métricas en función del número de observaciones evaluadas solo en el conjunto de supernovas de tipo Ia. Cada métrica es obtenida mediante el promedio de los 5 k-folds cuya desviación estándar se representa como un área sombreada. Se comparan los resultados para el modelo ALerCE-P, **SNeF all** (modelo entrenado con distintos tipos de supernovas) y **SNeF SNIa** (modelo entrenado solo con supernovas de tipo SNIa)

## 4.2. Predicciones

A pesar de que las métricas utilizadas muestran el comportamiento de las predicciones en un horizonte de predicción de 7 días, el modelo propuesto es capaz de predecir en horizontes

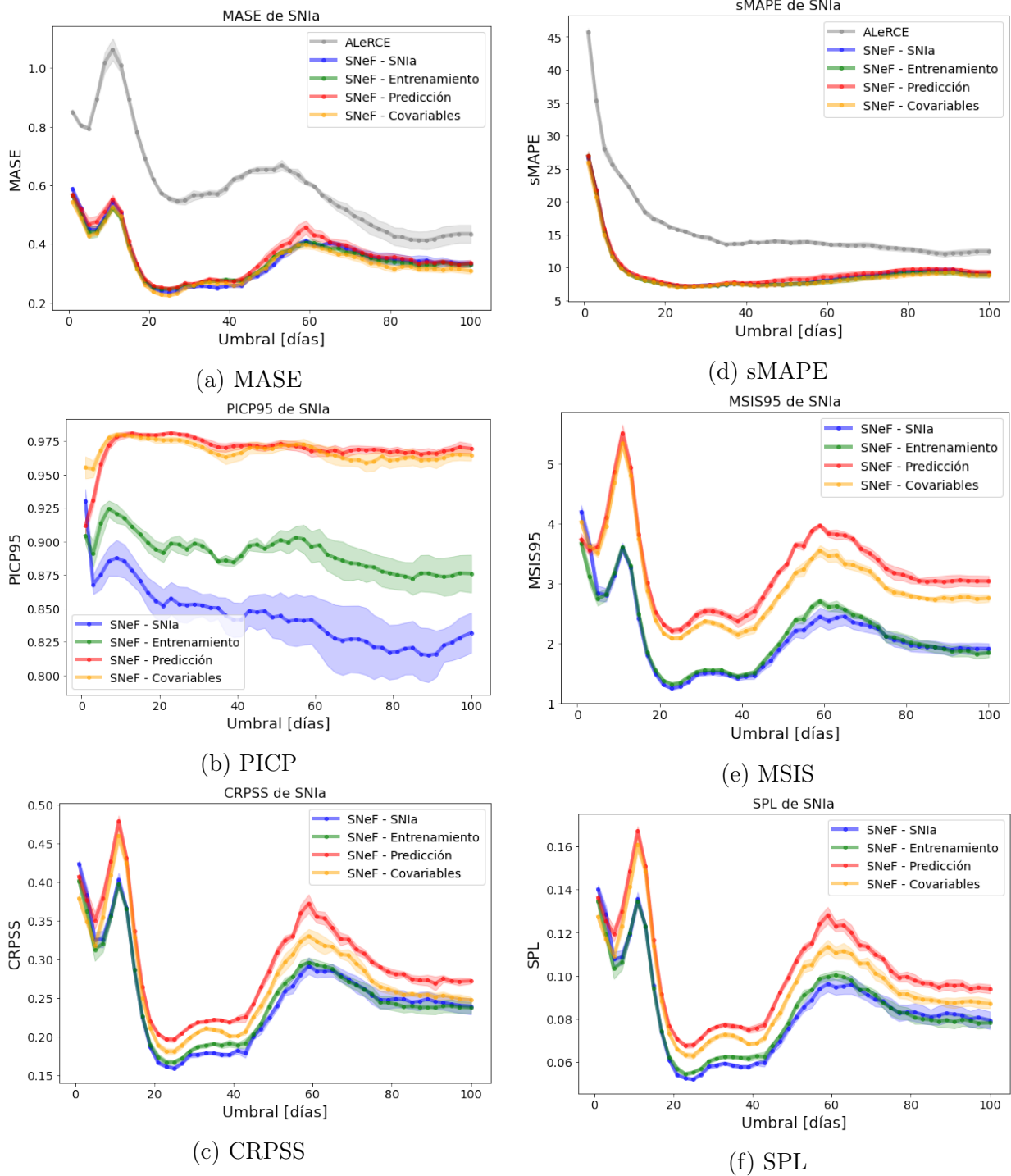


Figura 4.2: Conjunto de métricas en función del umbral de días evaluadas solo en el conjunto de supernovas de tipo Ia. Cada métrica es obtenida mediante el promedio de los 5 k-folds cuya desviación estándar se representa como un área sombreada. Se comparan los resultados para el modelo ALerCE-P, **SNeF all** (modelo entrenado con distintos tipos de supernovas) y **SNeF SNIa** (modelo entrenado solo con supernovas de tipo SNIa)

de predicción variados. En esta sección se presentan algunos ejemplos de predicciones para cada tipo de Supernovas, mostrando casos donde las predicciones siguen el comportamiento

de las supernovas y casos donde no lo siguen.

### 4.2.1. Predicciones con una observación

En astronomía existe el concepto de clasificación temprana [1, 2]. En el cual se clasifica el tipo de supernova correspondiente a la primera observación. En este caso, se busca estudiar el concepto de **predicción temprana**. Este consiste en generar una predicción basado solo en la primera observación realizada. Para realizar esta estimación se debe conocer la información del brillo (en magnitud o flujo), la banda a la cual corresponde y al tipo de supernova. En caso de no conocer el tipo de supernova, es posible realizar una estrategia de “qué pasaría sí”. La anterior se refiere a que se pueden realizar tantas predicciones como clases existan en el conjunto de datos.

En la Fig 4.4 se observa la predicción generada, a partir de un solo punto, de 6 curvas de luz **distintas**. Cada una de estas curvas puede estar en la banda  $r$  (representada por el color rojo) o en la banda  $g$  (representada por el color verde). Esta figura compara las predicciones del modelo ALerCE-P y el modelo SNeF - Covariables con la curva ajustada.

A partir de estos ejemplos se observa que la mediana del intervalo de predicción, mayoritariamente, no se acerca al comportamiento real observado de la curva de luz. A pesar de lo anterior, los intervalos de predicción obtenidos si logran capturar la mayoría de estos puntos, permitiendo obtener un límite superior e inferior coherente con los valores futuros. Por otro lado es interesante notar que las predicciones futuras tienen el comportamiento esperado de una curva de luz, es decir, se observa un periodo de crecimiento y posterior decaimiento.

Si bien es cierto, en general la predicción (mediana) no logra tener un buen desempeño, existen algunos casos que logra realizar un correcto modelamiento. (Ver esquina inferior derecha de la Fig C.2). Además, como se mostró anteriormente (Ver Fig 4.2a y 4.2d, 4.1a y 4.1d), incluso con un solo punto las predicciones superan a ALerCE y al modelo *naive*.

### 4.2.2. Ejemplo de predicción para una SNIa

En la Figura 4.4 se observa un ejemplo de la predicción de una supernova de tipo Ia según distintos números de puntos. A partir de esta se observa que el tamaño del intervalo de predicción tiende a disminuir a medida que se tienen más puntos. Es importante notar que en la Figura 4.1e se observa que el valor del MSIS es alto cuando se tienen pocas observaciones, pero este disminuye a medida que se tienen más. Luego de 15 observaciones, el valor del intervalo de predicción vuelve a aumentar, lo cual podría deberse a los factores de penalización y/o problemas en la escalamiento de la métrica.

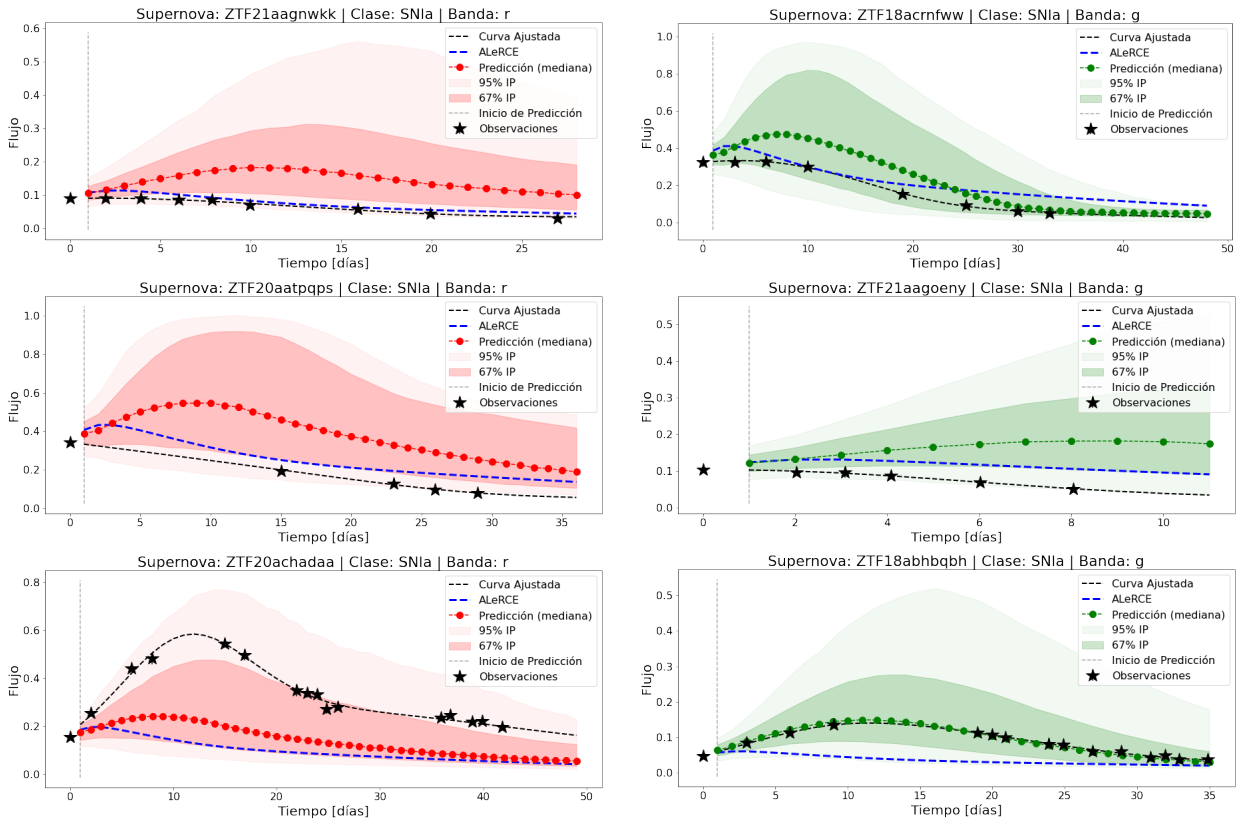


Figura 4.3: Ejemplos de predicciones de 6 curvas de luz **distintas** cuando se conoce un solo punto. En esta figura se muestra la comparación entre la curva ajustada (utilizada para evaluar), la predicción propuesta por ALeRCE y la predicción del modelo SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (Inicio de Predicción), donde el resto de observaciones solo son referenciales para comparar las predicciones

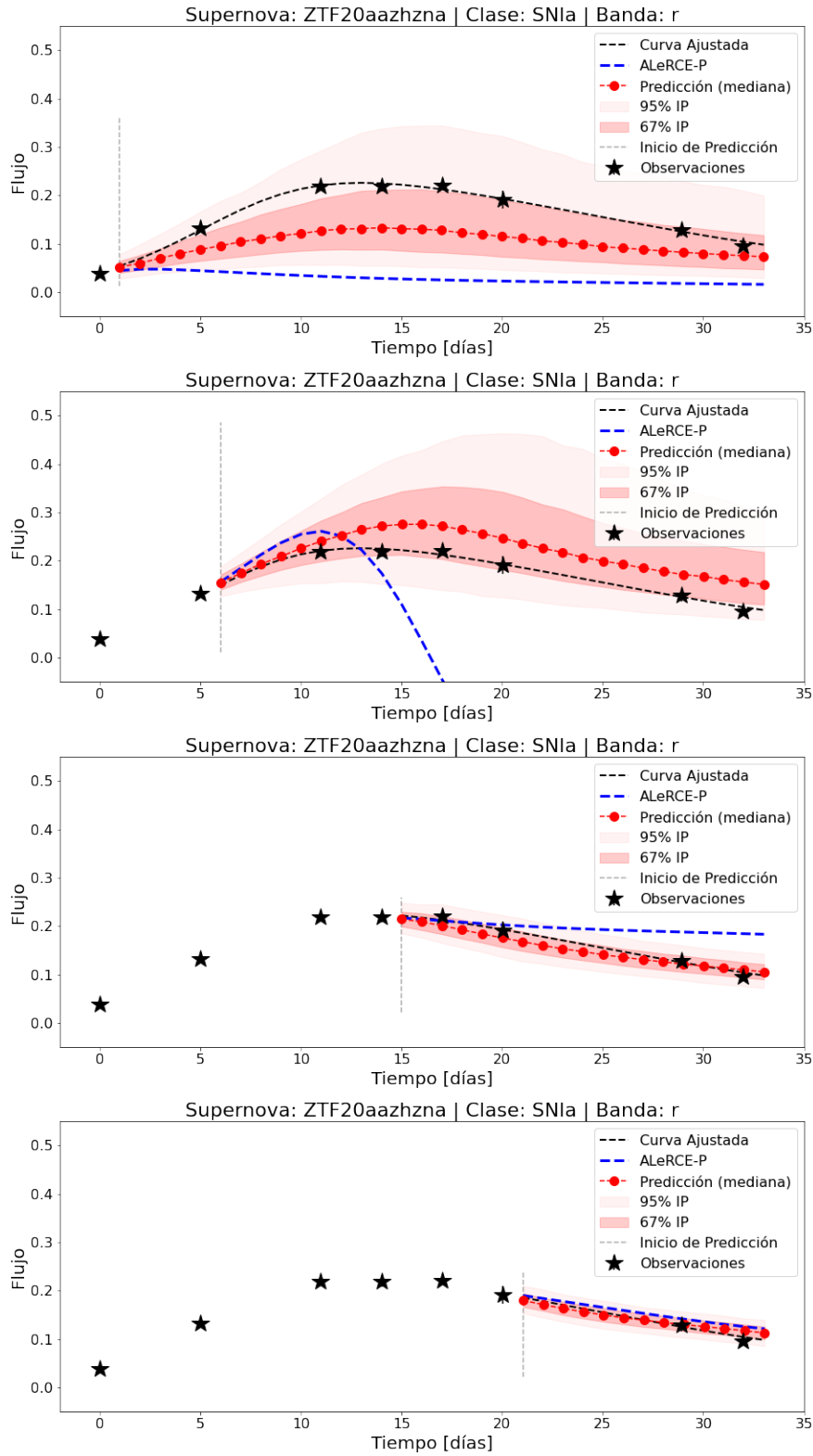


Figura 4.4: Ejemplo de predicciones de una misma curva de luz pero con distinta cantidad de puntos como entrada. Se comparan las predicciones después de 1, 2, 4 y 6 puntos, respectivamente. También se muestra la predicción realizada por el modelo ALeRCE-P.

## 5. Conclusión

En este trabajo se cumplieron todos los objetivos específicos propuestos. Se desarrolló un predictor de curvas de luz de supernovas basado en redes neuronales recurrentes, denominado SNeF. Este corresponde a uno de los primeros modelos basados en técnicas de aprendizaje profundo que resuelve la tarea de predecir curvas de luz de supernovas de distintos tipos. Además, es importante notar que este predictor se construyó y evaluó en base a datos reales de curvas de luz de distintas clases de la base de datos provista por ALeRCE-ZTF.

El modelo propuesto permite realizar predicciones univariadas de curvas de luz de supernovas a un paso de 1 día sobre horizontes de predicción de distinto largo. Además es capaz de realizar predicciones con pocos datos: antes de alcanzar el momento de máximo brillo e incluso, con una sola observación. Ejemplos visuales de lo descrito anteriormente se observan en las figuras presentadas en la Sección 4.2 y Anexo C.

Por otro lado, dado que el modelo SNeF realiza predicciones probabilísticas es posible estimar los límites mínimos y máximos que los valores futuros puede alcanzar con una confianza cercana al 98%, es decir, estos límites capturan la mayoría de las observaciones futuras de las supernovas. Lo anterior permitiría a los astrónomos calibrar sus telescopios para capturar de mejor forma el brillo de las supernovas.

Uno de los objetivos específicos del trabajo corresponde a idear e implementar estrategias que consideren el error de observación y el modelo paramétrico de supernovas para mejorar el entrenamiento, la predicción y validación del modelo.

Como se mostró en la Sección 4.1, perturbar los valores de entrada del modelo, permitió mejorar la confianza de los intervalos de predicción a un costo de aumentar su tamaño. A pesar de esto, los ejemplos visuales (Ver Sección 4.2 y Anexo C) muestran que estos intervalos, para la mayoría de los casos, son acotados.

Por otro lado, se mostró la importancia que tiene agregar covariables al modelo para mejorar los resultados. En este caso, a partir del uso de la primera observación (flujo y distancia en tiempo) como covariables, el desempeño del modelo mejoró disminuyendo el tamaño de su intervalo de predicción. A pesar de que no se exploraron más covariables, esto abre una oportunidad de trabajo futuro que será señalado en la siguiente sección.

Finalmente, dado que este es uno de los primeros trabajos que se realizan sobre predicción de supernovas, se planteó una metodología de evaluación para comparar el desempeño del modelo bajo distintas condiciones. Esta estrategia permite evaluar el desempeño del modelo incluso cuando no existen observaciones con las cuáles compararse.

## 5.1. Trabajo Futuro

Esta tesis es uno de los primeros estudios sobre la tarea de predicción de curvas de luz mediante técnicas de aprendizaje de máquinas. Dado lo anterior, existe una gran cantidad de análisis que se podrían realizar en base a los resultados de la presente tesis. A continuación se presentan propuestas que fueron abordadas parcialmente en algún momento del desarrollo de la tesis, pero se espera que podrían tener un alto impacto en el análisis de curvas de luz de supernovas.

### 5.1.1. Modelos base para comparar

En el mejor de los esfuerzos no se encontró literatura asociada a la predicción de curvas de luz de supernovas. Es por esto que para comparar los modelos propuestos se requiere también trabajar con modelos simples que permitan comparar el desempeño de las métricas.

En base a lo anterior, no existe un modelo probabilístico de predicción de supernovas con el cuál se pudiese comparar los resultados del método propuesto. Se podría utilizar un modelo base que, al igual que el método de ALERCE (Ver Sección 3.4 ), utilice el modelo paramétrico de supernova pero obteniendo un conjunto de parámetros a partir de la técnica MCMC. A partir de lo anterior, sería posible generar predicciones y obtener barras de error asociadas a esta. El problema radica en que realizar este procedimiento de forma iterativa sería costoso computacionalmente.

Por otro lado, también se propone la utilización de técnicas más simples, como lo son regresiones lineales, regresiones polinómicas, entre otras. A partir de esto, sería posible robustecer el análisis del desempeño del modelo probabilístico. En [49, 102] se presenta una serie de líneas base que podrían ser utilizadas a modo de inspiración para encontrar otros modelos.

### 5.1.2. Horizonte y pasos de predicción

En esta tesis se realizaron estudios preliminares para determinar la influencia de la diferencia de tiempo como entrada en el resultado de los modelos. En primera instancia se observó que el número de pasos para alcanzar el horizonte de predicción afectaba el desempeño del modelo. Lo anterior significa que al predecir el instante de tiempo de  $t + 3$ , no se obtienen resultados consistentes ya sea prediciendo a un paso de un día  $t, t + 1, t + 2, t + 3$  o a un paso de 3 días  $t, t + 3$ .

Preliminarmente esto indicaría que el modelo solo podría predecir a un paso de tiempo fijo (1 día). Dado lo anterior se espera estudiar en trabajos futuros el comportamiento de las predicciones según el paso de predicción escogido. Sería deseable que un futuro modelo pudiese generalizar las predicciones independientes del paso de predicción.

### 5.1.3. Búsqueda de hiperparámetros

Dado el alto costo computacional, no se realizó una búsqueda exhaustiva de los mejores hiperparámetros para cada uno de las variaciones de los modelos propuestos. En base a lo anterior, se esperaría que al realizar esta búsqueda se encuentren hiperparámetros que permitan mejorar el desempeño del modelo.

### 5.1.4. Utilización de más covariables

Como se mostró en la tesis, el rol de las covariables en la tarea de predicción es clave para mejorar la precisión de estas [75]. Es por esto, que se propone explorar covariables que puedan ser conocidas en todo momento (pasado y futuro). Algunas de estas podrían ser:

- Predicciones de otros modelos: Se podrían utilizar modelos simples de predicción como covariables para mejorar el desempeño del modelo. Un ejemplo de esto, podría ser utilizando el modelo ALerCE-P.
- Información de otras bandas: En este trabajo se abordó el enfoque univariado, es decir, estudio de predicciones por cada una de las bandas. A pesar de lo anterior, como se ha mostrado en tareas de clasificación, agregar la información de ambas bandas permitiría mejorar el desempeño de la tarea. Es por esto, que se propone buscar una estrategia para aprovechar la información de todas las bandas existentes. Se espera que la información entre bandas permita disminuir los periodos de tiempo sin observaciones, además de limitar las predicciones a los valores de las bandas.
- Metadatos astronómicos: Utilizar metadatos astronómicos podría mejorar el desempeño del modelo. Entre estos podría considerarse el uso del *redshift*.

### 5.1.5. Irregularidad en el muestreo de los datos

Como se mencionó en esta tesis, existen diversas formas de trabajar la irregularidad en el muestreo de los datos. En este trabajo se realizó una imputación lineal simple entre las observaciones conocidas (Ver Sección 3.4).

Si bien es cierto que la imputación lineal es una metodología simple, esta podría agregar sesgos a los datos que se traduciría en un peor desempeño del modelo. Dado lo anterior, se propone explorar otras formas de abordar la irregularidad en el muestreo de los datos.

Una opción es la modificación de la celda recurrente. En [83] se agrega un factor de decaimiento según el tiempo transcurrido entre dos observaciones. Este factor de decaimiento se incluye directamente en la celda de la red neuronal recurrente.

Por otro lado, en [112] se propone generalizar la obtención de los estados ocultos discretos de la RNN mediante la utilización de ecuaciones diferenciales ordinarias. A partir de lo anterior, se busca obtener una representación continua y diferenciable de los estados ocultos, lo que permitiría incluir características temporales.

Finalmente, otra metodología que podría ser utilizada para superar la irregularidad en el muestreo de los datos, es imputar los datos según la salida del modelo. Se espera que el



modelo pueda auto completarse, donde el procedimiento de imputación y predicción serían similares.

### **5.1.6. Complementar tareas de clasificación o detección de anomalías**

Si bien es cierto, cada una de estas tareas trae consigo sus propias complejidades, se propone explorar si las predicciones generadas por el modelo propuesto serían útiles para mejorar las tareas de clasificación y detección de anomalías.

#### **5.1.6.1. Mejoras en tareas de clasificación**

Para mejorar la tarea de clasificación se propone predecir el comportamiento futuro de una supernova a partir de una cierta cantidad de observaciones dada. En base a esto, como se tendría una estimación de la predicción y de su incertidumbre, esta podría ser ingresada al modelo de clasificación como una observación cuyo ruido es proporcional a la incertidumbre asociada al intervalo de predicción.

Según lo mostrado en este trabajo, a partir de pocos puntos es posible generar una predicción completa de la curva de luz, lo que podría ser ingresado al clasificador. De esta forma, se esperaría que el clasificador tuviera información suficiente para mejorar su desempeño.

#### **5.1.6.2. Tareas de detección de anomalías**

La tarea de predicción se podría aprovechar para detectar curvas de luz anómalas [12,13]. Por ejemplo en [7, 11] se propone calcular la diferencia entre la observación de una fuente luminosa y la predicción realizada por el predictor (SNeF). Esta diferencia podría alertar la existencia de un evento anómalo.

En este caso, suponiendo que los intervalos de predicción son confiables, se esperaría que si la observación futura no cae dentro del intervalo de predicción entonces se trataría de una observación y/o evento anómalo.

### **5.1.7. Robustecer modelo de predicción**

En imágenes se suele utilizar el concepto de ataque adversario, el cual a grandes rasgos significa que a partir de perturbaciones imperceptibles para el ser humano, se puede afectar el desempeño de un clasificador de imágenes.

En [113] se explica como se podría realizar un ataque adversario a la tarea de predicción al utilizar redes neuronales recurrentes. Además, en este mismo trabajo se explica la forma de robustecer el modelo ante este tipo de ataques, permitiendo mejorar el desempeño. Lo anterior permitiría mejorar el desempeño del modelo y volverlo robusto ante estos ataques.

### 5.1.8. Utilización de arquitectura *Transformers*

La arquitectura *Transformer* ha alcanzado resultados prometedores en el área de astronomía [19]. Por esto se propone explorar la utilización de esta arquitectura en futuros trabajos. A partir de esta arquitectura, se esperaría lidiar de mejor forma la irregularidad en el muestreo de los datos en comparación al modelo propuesto.

### 5.1.9. Estimación de nuevos parámetros

En el presente trabajo se asumió que la distribución de la predicción correspondía a una Gaussiana. A partir de esta se estimaron dos parámetros: media y desviación estándar. Esto se realizó siguiendo el esquema mostrado en [74]. A pesar de que con este enfoque los resultados alcanzados lograron predecir el comportamiento futuro de la curva de luz de supernova, este podría ser mejorado a través de la estimación de nuevos parámetros y distribuciones. En [63, 72] proponen metodologías para estimar parámetros y distribuciones de salidas diferentes a las trabajadas en esta tesis.

En este caso en particular, se podrían estimar los parámetros del modelo paramétricos de supernovas (SPM) utilizando como distribuciones las propuestas en las restricciones de estos parámetros (ver [19]). De esta forma se podría capturar de mejor forma la dinámica de las predicciones.

Estimar nuevos parámetros podría aumentar la varianza en las predicciones, pues dependerán de más variables. Además, dependiendo de las distribuciones que se utilicen, podría aumentar el costo computacional de los experimentos.

# Bibliografía

- [1] F. Förster, G. Cabrera-Vives, E. Castillo-Navarrete, P. A. Estévez, P. Sánchez-Sáez, J. Arredondo, *et al.*, “The automatic learning for the rapid classification of events (alerce) alert broker,” *The Astronomical Journal*, vol. 161, p. 242, 5 2021.
- [2] R. Carrasco-Davis, E. Reyes, C. Valenzuela, F. Förster, P. A. Estévez, *et al.*, “Alert classification for the alerce broker system: The real-time stamp classifier,” *The Astronomical Journal*, vol. 162, p. 231, 12 2021.
- [3] N. Sravan, D. Milisavljevic, J. M. Reynolds, G. Lentner, and M. Linvill, “Real-time, value-driven data augmentation in the era of lsst,” *The Astrophysical Journal*, vol. 893, p. 127, 4 2020.
- [4] A. Grigorievskiy, M. M. A. Akusok, E. Eirola, and A. Lendasse, “Forecasting the outbursts of the photometry light curve of star v363 lyr,” in *International Work Conference on Time Series Analysis 2014, Granada*, pp. 520–531, 2014.
- [5] E. D. Feigelson, G. J. Babu, and G. A. Caceres, “Autoregressive times series methods for time domain astronomy,” *Frontiers in Physics*, vol. 6, 2018.
- [6] N. Chakraborty, “Testing rnn-lstm forecasting with simulated astronomical lightcurves,” *Preprints 2019*, 2019070241. <https://doi.org/10.20944/preprints201907.0241.v1>, 7 2019.
- [7] R. Zhang and Q. Zou, “Time series prediction and anomaly detection of light curve using lstm neural network,” in *Journal of Physics: Conference Series*, vol. 1061, p. 012012, IOP Publishing, 2018.
- [8] X. Huang, L. Peng, C. Lu, J. Bi, and H. Yuan, “A deep learning approach to large-scale light curve prediction and real-time anomaly detection with grubbs criterion,” in *2020 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1–6, IEEE, 2020.
- [9] Y. Tachibana, M. J. Graham, N. Kawai, S. G. Djorgovski, A. J. Drake, A. A. Mahabal, and D. Stern, “Deep modeling of quasar variability,” *The Astrophysical Journal*, vol. 903, p. 54, 3 2020.
- [10] I. R. Goumiri, A. M. Dunton, A. L. Muyskens, B. W. Priest, and R. E. Armstrong, “Light curve completion and forecasting using fast and scalable gaussian processes (muygps),” *arXiv preprint arXiv:2208.14592*, 2022.

- [11] D. Muthukrishna, K. S. Mandel, M. Lochner, S. Webb, and G. Narayan, “Real-time detection of anomalies in large-scale transient surveys,” *Monthly Notices of the Royal Astronomical Society*, vol. 517, pp. 393–419, 10 2022.
- [12] P. Sánchez-Sáez, H. Lira, L. Martí, N. Sánchez-Pi, J. Arredondo, F. E. Bauer, *et al.*, “Searching for changing-state agns in massive data sets. i. applying deep learning and anomaly-detection techniques to find agns with anomalous variability behaviors,” *The Astronomical Journal*, vol. 162, p. 206, 11 2021.
- [13] M. Perez-Carrasco, G. Cabrera-Vives, L. Hernandez-García, F. Förster, *et al.*, “Alert classification for the alerce broker system: The anomaly detector,” *The Astronomical Journal*, vol. 166, p. 151, 10 2023.
- [14] P. Sánchez-Sáez, I. Reyes, C. Valenzuela, F. Förster, S. Eyheramendy, F. Elorrieta, *et al.*, “Alert classification for the alerce broker system: The light curve classifier,” *The Astronomical Journal*, vol. 161, p. 141, 3 2021.
- [15] G. Bazin, N. Palanque-Delabrouille, J. Rich, V. Ruhlmann-Kleider, E. Aubourg, L. L. Guillou, P. Astier, *et al.*, “The core-collapse rate from the supernova legacy survey,” *Astronomy and Astrophysics*, vol. 499, pp. 653–660, 6 2009.
- [16] J. Newling, M. Varughese, B. Bassett, H. Campbell, R. Hlozek, M. Kunz, H. Lampeitl, B. Martin, *et al.*, “Statistical classification techniques for photometric supernova typing,” *Monthly Notices of the Royal Astronomical Society*, vol. 414, pp. 1987–2004, 7 2011.
- [17] N. V. Karpenka, F. Feroz, and M. P. Hobson, “A simple and robust method for automated photometric classification of supernovae using neural networks,” *Monthly Notices of the Royal Astronomical Society*, vol. 429, pp. 1278–1285, 2 2013.
- [18] V. A. Villar, E. Berger, G. Miller, R. Chornock, A. Rest, D. O. Jones, *et al.*, “Supernova photometric classification pipelines trained on spectroscopically classified supernovae from the pan-starrs1 medium-deep survey,” *The Astrophysical Journal*, vol. 884, p. 83, 10 2019.
- [19] Óscar Pimentel, P. A. Estévez, and F. Förster, “Deep attention-based supernovae classification of multi-band light-curves,” *The Astronomical Journal*, vol. 165, p. 18, 1 2022.
- [20] S. Vaughan, “Random time series in astronomy,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110549, 2013.
- [21] F. Förster, J. C. Maureira, J. S. Martín, M. Hamuy, J. Martínez, P. Huijse, *et al.*, “The high cadence transient survey (hits). i. survey design and supernova shock breakout constraints,” *The Astrophysical Journal*, vol. 832, p. 155, 11 2016.
- [22] R. Dekany, R. M. Smith, R. Riddle, M. Feeney, M. Porter, D. Hale, *et al.*, “The zwicky transient facility: Observing system,” *Publications of the Astronomical Society of the Pacific*, vol. 132, p. 038001, 3 2020.

- [23] Željko Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, *et al.*, “Lsst: From science drivers to reference design and anticipated data products,” *The Astrophysical Journal*, vol. 873, p. 111, 3 2019.
- [24] P. Huijse, P. A. Estévez, F. Förster, S. F. Daniel, A. J. Connolly, P. Protopapas, R. Carrasco, and J. C. Príncipe, “Robust period estimation using mutual information for multiband light curves in the synoptic survey era,” *The Astrophysical Journal Supplement Series*, vol. 236, p. 12, 5 2018.
- [25] R. Hyndman, A. Koehler, K. Ord, and R. Snyder, *Forecasting with Exponential Smoothing*. Springer Berlin Heidelberg, 2008.
- [26] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Springer International Publishing, 2016.
- [27] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, “A review of irregular time series data handling with gated recurrent neural networks,” *Neurocomputing*, vol. 441, pp. 161–178, 6 2021.
- [28] S. R. Beeram and S. Kuchibhotla, “Time series analysis on univariate and multivariate variables: A comprehensive survey,” vol. 134, pp. 119–126, Springer Science and Business Media Deutschland GmbH, 2021.
- [29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.
- [30] A. M. Schäfer and H. G. Zimmermann, “Recurrent neural networks are universal approximators,” vol. 4131 LNCS - I, pp. 632–640, Springer Verlag, 2006.
- [31] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, pp. 179–211, 4 1990.
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.
- [33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 12 2014.
- [34] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [35] C. M. Bishop, “Training with noise is equivalent to tikhonov regularization,” *Neural Computation*, vol. 7, pp. 108–116, 1 1995.
- [36] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.
- [37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

- [38] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” pp. 1027–1035, NIPS, 2016.
- [39] T. Moon, H. Choi, H. Lee, and I. Song, “Rnndrop: A novel dropout for rnns in asr,” *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 65–70, 2015.
- [40] S. Semeniuta, A. Severyn, and E. Barth, “Recurrent dropout without memory loss,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, (Osaka, Japan), pp. 1757–1766, The COLING 2016 Organizing Committee, Dec. 2016.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al.*, “Attention is all you need,” vol. 30, Curran Associates, Inc., 2017.
- [42] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [43] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, “An empirical comparison of machine learning models for time series forecasting,” *Econometric Reviews*, vol. 29, pp. 594–621, 2010.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [46] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [47] R. J. Hyndman, “A brief history of forecasting competitions,” *International Journal of Forecasting*, vol. 36, pp. 7–14, 1 2020.
- [48] B. Lim and S. Zohren, “Time-series forecasting with deep learning: a survey,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, p. 20200209, 4 2021.
- [49] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “The m4 competition: 100,000 time series and 61 forecasting methods,” *International Journal of Forecasting*, vol. 36, pp. 54–74, 1 2020.
- [50] K. Benidis, S. S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, C. Turkmen, *et al.*, “Deep learning for time series forecasting: Tutorial and literature survey,” *ACM Computing Surveys*, 5 2022.
- [51] A. Borovykh, S. Bohte, and C. W. Oosterlee, “Conditional time series forecasting with convolutional neural networks,” *arXiv preprint arXiv:1703.04691*, 2017.

- [52] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.
- [53] C. Fan, Y. Zhang, Y. Pan, X. Li, C. Zhang, R. Yuan, D. Wu, W. Wang, J. Pei, and H. Huang, “Multi-horizon time series forecasting with temporal attention learning,” in *Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining*, pp. 2527–2535, 2019.
- [54] B. Lim, S. Arik, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, pp. 1748–1764, 10 2021.
- [55] M. Al-Gabalawy, N. S. Hosny, and A. R. Adly, “Probabilistic forecasting for energy time series considering uncertainties based on deep learning algorithms,” *Electric Power Systems Research*, vol. 196, p. 107216, 7 2021.
- [56] E. Ng, Z. Wang, H. Chen, S. Yang, and S. Smyl, “Orbit: probabilistic forecast with exponential smoothing,” *arXiv preprint arXiv:2004.08492*, 2020.
- [57] H. G. Zimmermann, C. Tietz, and R. Grothmann, “Forecasting with recurrent neural networks: 12 tricks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7700 LECTURE NO, pp. 687–707, 2012.
- [58] T. Yoon, Y. Park, E. K. Ryu, and Y. Wang, “Robust probabilistic time series forecasting,” in *International Conference on Artificial Intelligence and Statistics*, pp. 1336–1358, PMLR, 2022.
- [59] B. Naul, J. S. Bloom, F. Pérez, and S. V. D. Walt, “A recurrent neural network for classification of unevenly sampled variable stars,” *Nature Astronomy*, vol. 2, pp. 151–155, 2 2018.
- [60] C. R. Rivero, J. Pucheta, P. Otaño, A. D. Orjuela-Cañon, D. Patiño, L. Franco, *et al.*, “Time series forecasting using recurrent neural networks modified by bayesian inference in the learning process,” in *2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI)*, pp. 1–6, IEEE, 2019.
- [61] M. Mvubu, E. Kabuga, C. Plitz, B. Bah, R. Becker, and H. G. Zimmermann, “On error correction neural networks for economic forecasting,” in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2020.
- [62] S. Pal, L. Ma, Y. Zhang, and M. Coates, “Rnn with particle flow for probabilistic spatio-temporal forecasting,” in *International Conference on Machine Learning*, pp. 8336–8348, PMLR, 2021.
- [63] A. Gouttes, K. Rasul, M. Koren, J. Stephan, *et al.*, “Probabilistic time series forecasting with implicit quantile networks,” *arXiv preprint arXiv:2107.03743*, 2021.
- [64] K. Stankeviciute, A. M Alaa, and M. van der Schaar, “Conformal time-series forecasting,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 6216–6228, 2021.

- [65] Y.-S. Lim and D. Gorse, “Deep probabilistic modelling of price movements for high-frequency trading,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.
- [66] J. Bottieau, L. Hubert, Z. D. Grève, F. Vallée, and J. F. Toubeau, “Very-short-term probabilistic forecasting for a risk-aware participation in the single price imbalance settlement,” *IEEE Transactions on Power Systems*, vol. 35, pp. 1218–1230, 3 2020.
- [67] H. Du, S. Du, and W. Li, “Probabilistic time series forecasting with deep non-linear state space models,” *CAAI Transactions on Intelligence Technology*, 4 2022.
- [68] N. Nguyen and B. Quanz, “Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 9117–9125, 2021.
- [69] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, “Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting,” in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868, PMLR, 18–24 Jul 2021.
- [70] T. Gneiting and A. E. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *Journal of the American Statistical Association*, vol. 102, pp. 359–378, 3 2007.
- [71] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, “Deep state space models for time series forecasting,” *Advances in neural information processing systems*, vol. 31, 2018.
- [72] L. Li, J. Yan, X. Yang, and Y. Jin, “Learning interpretable deep state space model for probabilistic time series forecasting,” vol. 2019-August, pp. 2901–2908, International Joint Conferences on Artificial Intelligence, 2019.
- [73] R. Wen and K. Torkkola, “Deep generative quantile-copula models for probabilistic forecasting,” *arXiv preprint arXiv:1907.10697*, 2019.
- [74] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “Deepar: Probabilistic forecasting with autoregressive recurrent networks,” *International Journal of Forecasting*, vol. 36, pp. 1181–1191, 7 2020.
- [75] S. Jung, K.-M. Kim, H. Kwak, and Y.-J. Park, “A worrying analysis of probabilistic time-series models for sales forecasting,” in *“I Can’t Believe It’s Not Better!”NeurIPS 2020 workshop*.
- [76] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, “Doctor ai: Predicting clinical events via recurrent neural networks,” in *Machine learning for healthcare conference*, pp. 301–318, PMLR, 2016.
- [77] D. Neil, M. Pfeiffer, and S.-C. Liu, “Phased lstm: Accelerating recurrent network training for long or event-based sequences,” in *Advances In Neural Information Processing Systems*, pp. 3882–3890, 2016.
- [78] T. Pham, T. Tran, D. Phung, and S. Venkatesh, “Deepcare: A deep dynamic memory model for predictive medicine,” *Lecture Notes in Computer Science*, 04 2016.



- [79] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, “Patient subtyping via time-aware lstm networks,” KDD ’17, (New York, NY, USA), p. 65–74, Association for Computing Machinery, 2017.
- [80] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, 04 2018.
- [81] Y. Lee, E. Jun, and H.-I. Suk, “Multi-view integration learning for irregularly-sampled clinical time series,” *arXiv preprint arXiv:2101.09986*, 2021.
- [82] Z. C. Lipton, D. C. Kale, R. Wetzel, *et al.*, “Modeling missing data in clinical time series with rnns,” *Machine Learning for Healthcare*, vol. 56, pp. 253–270, 2016.
- [83] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, pp. 1–12, 12 2018.
- [84] Y. Bengio and F. Gingras, “Recurrent neural networks for missing or asynchronous data,” *Advances in neural information processing systems*, vol. 8, 1995.
- [85] M. Lepot, J.-B. Aubin, and F. Clemens, “Interpolation in time series: An introductory overview of existing methods, their performance criteria and uncertainty assessment,” *Water*, vol. 9, p. 796, 10 2017.
- [86] M. Nguyen, N. Sun, D. C. Alexander, J. Feng, and B. T. Yeo, “Modeling alzheimer’s disease progression using deep recurrent neural networks,” in *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*, pp. 1–4, IEEE, 2018.
- [87] S. N. Shukla and B. M. Marlin, “Interpolation-prediction networks for irregularly sampled time series,” *arXiv preprint arXiv:1909.07782*, 2019.
- [88] S. Eyheramendy, F. Elorrieta, and W. Palma, “An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves,” *Monthly Notices of the Royal Astronomical Society*, vol. 481, pp. 4311–4322, 12 2018.
- [89] F. Elorrieta, S. Eyheramendy, and W. Palma, “Discrete-time autoregressive model for unequally spaced time-series observations,” *Astronomy and Astrophysics*, vol. 627, p. A120, 7 2019.
- [90] N. Metropolis and S. Ulam, “The monte carlo method,” *Journal of the American Statistical Association*, vol. 44, p. 335, 9 1949.
- [91] L. Rabiner and B. Juang, “An introduction to hidden markov models,” *iee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.
- [92] C. P. Robert and S. Richardson, “Markov chain monte carlo methods,” *Discretization and MCMC Convergence Assessment*, pp. 1–25, 1998.
- [93] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, “emcee : The mcmc hammer,” *Publications of the Astronomical Society of the Pacific*, vol. 125, pp. 306–312, 3 2013.

- [94] J. Goodman and J. Weare, “Ensemble samplers with affine invariance,” *Communications in Applied Mathematics and Computational Science*, vol. 5, pp. 65–80, 2010.
- [95] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, pp. 679–688, 10 2006.
- [96] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, “Time series data augmentation for deep learning: A survey,” vol. 5, pp. 4653–4660, International Joint Conferences on Artificial Intelligence, 8 2021.
- [97] S. Makridakis, “Accuracy measures: theoretical and practical concerns,” *International Journal of Forecasting*, vol. 9, pp. 527–529, 12 1993.
- [98] T. Gneiting, F. Balabdaoui, and A. E. Raftery, “Probabilistic forecasts, calibration and sharpness,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, pp. 243–268, 4 2007.
- [99] J. F. Toubreau, J. Bottieau, Y. Wang, and F. Vallee, “Interpretable probabilistic forecasting of imbalances in renewable-dominated electricity systems,” *IEEE Transactions on Sustainable Energy*, vol. 13, pp. 1267–1277, 4 2022.
- [100] H. Zhong and L. Xu, “An all-batch loss for constructing prediction intervals,” *Applied Sciences*, vol. 11, p. 1728, 2 2021.
- [101] R. L. Winkler, “A decision-theoretic approach to interval estimation,” *Journal of the American Statistical Association*, vol. 67, p. 187, 3 1972.
- [102] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “M5 accuracy competition: Results, findings, and conclusions,” *International Journal of Forecasting*, vol. 38, pp. 1346–1364, 10 2022.
- [103] A. Jordan, F. Krüger, and S. Lerch, “Evaluating probabilistic forecasts with scoringrules,” *Journal of Statistical Software*, vol. 90, pp. 1–37, 8 2019.
- [104] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, and R. Vollgraf, “Multivariate probabilistic time series forecasting via conditioned normalizing flows,” *arXiv preprint arXiv:2002.06103*, 2020.
- [105] V. Le Guen and N. Thome, “Probabilistic Time Series Forecasting with Structured Shape and Temporal Diversity,” in *NeurIPS 2020*, (Vancouver, Canada), Dec. 2020.
- [106] B. K. Iwana and S. Uchida, “An empirical survey of data augmentation for time series classification with neural networks,” *PLOS ONE*, vol. 16, p. e0254841, 7 2021.
- [107] D. Muthukrishna, G. Narayan, K. S. Mandel, R. Biswas, *et al.*, “Rapid: Early classification of explosive transients using deep learning,” *Publications of the Astronomical Society of the Pacific*, vol. 131, p. 118002, 11 2019.
- [108] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.

- [109] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.
- [110] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*, pp. 1310–1318, PMLR, 2013.
- [111] D. Masters and C. Luschi, “Revisiting small batch training for deep neural networks,” *arXiv preprint arXiv:1804.07612*, 2018.
- [112] Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud, “Latent ordinary differential equations for irregularly-sampled time series,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
- [113] R. Dang-Nhu, G. Singh, P. Bielik, and M. Vechev, “Adversarial attacks on probabilistic autoregressive forecasting models,” in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 2356–2365, PMLR, 13–18 Jul 2020.

# Anexos

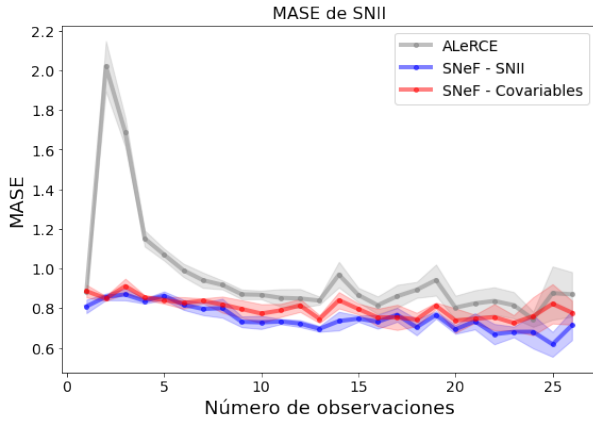
# Anexo A

## Métricas para SNII, SNIbc, SLSN

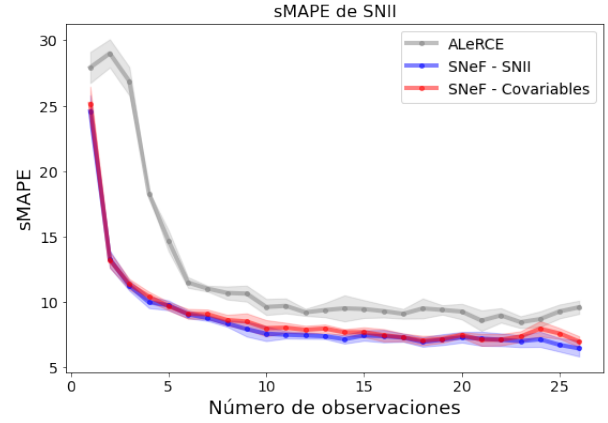
En esta sección se muestran las figuras asociadas a la comparación de las métricas de los distintos tipos de supernovas en base a los modelos SNeF - Tipo y SNeF - Covariables. Se muestra un conjunto de gráficos para cada tipo de evaluación propuesta: umbral de días (Figuras A.2, A.4, A.6) y número de observaciones (Figuras A.1, A.3, A.5)

SNeF - Tipo corresponde al modelo entrenado solo con un tipo de supernovas. Además, a diferencia de los resultados obtenidos en el cuerpo de la tesis, este modelo considera las covariables propuestas.

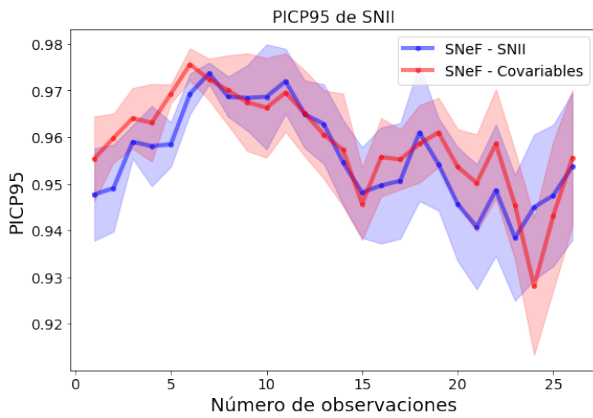
SNeF - Covariables corresponde al mismo modelo anterior, solo que entrenado con todos los tipos de luz de supernovas.



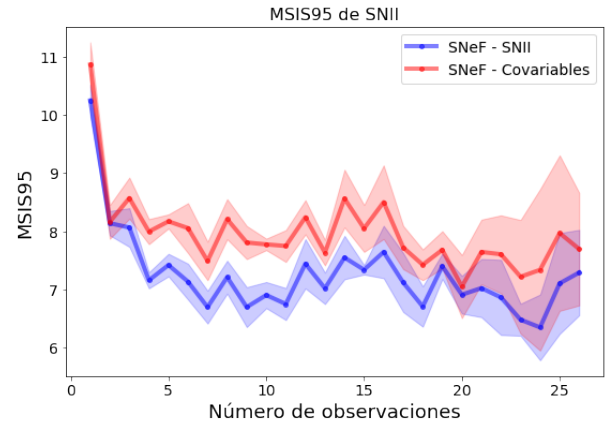
(a) MASE



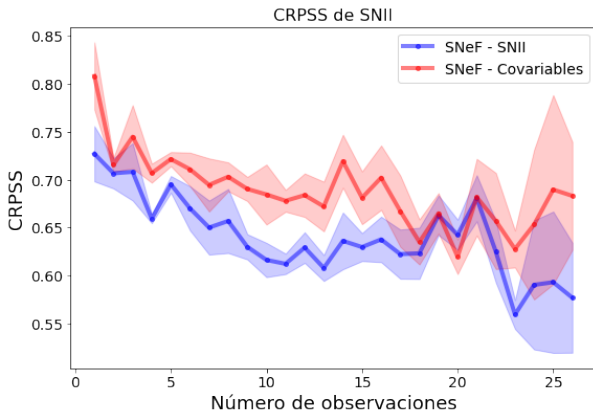
(d) sMAPE



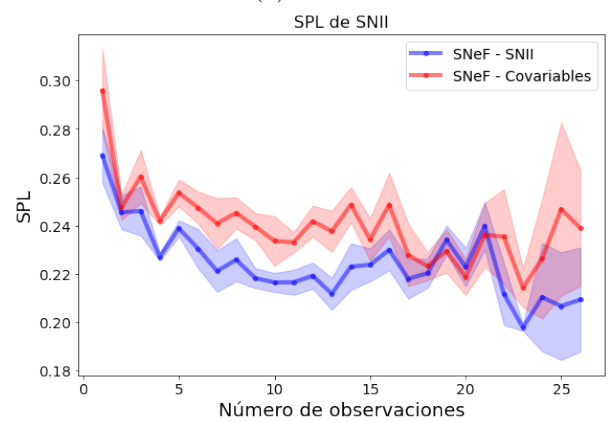
(b) PICP



(e) MSIS

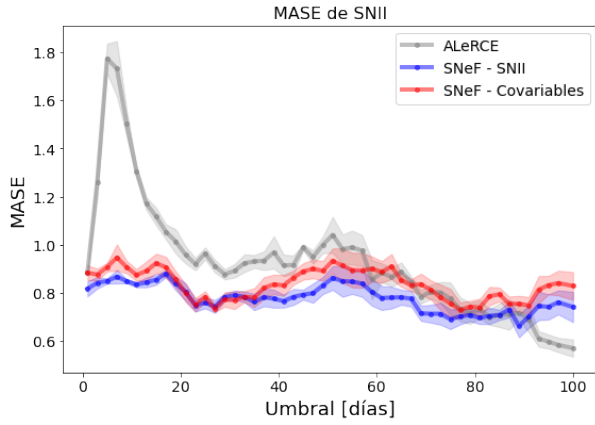


(c) CRPSS

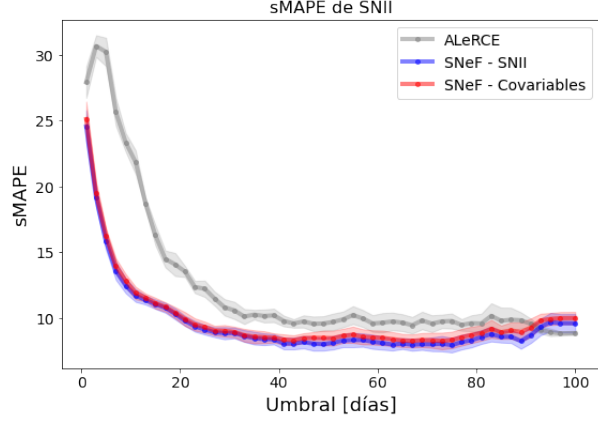


(f) SPL

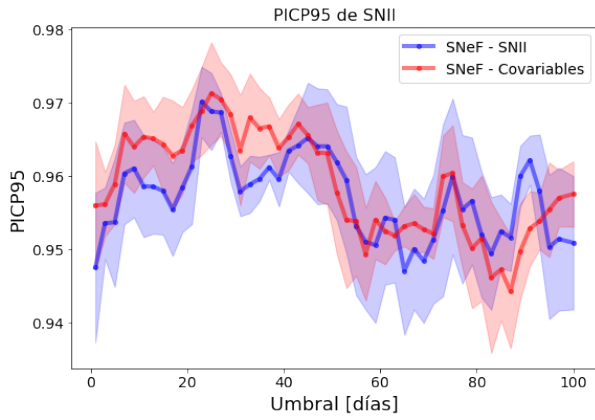
Figura A.1: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALeRCE-P, **SNeF-SNII** (modelo entrenado solo con supernovas de tipo SNII) y **SNeF - Covariables** (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



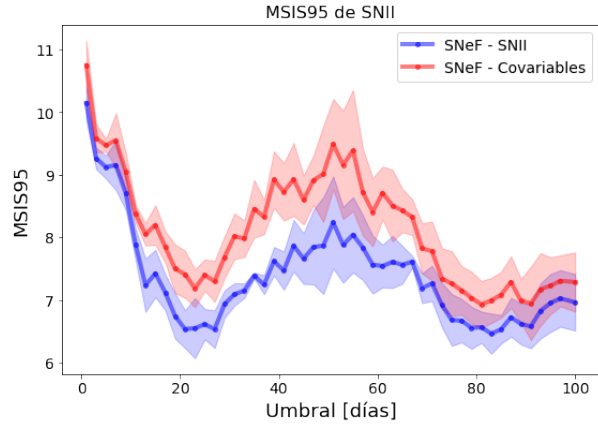
(a) MASE



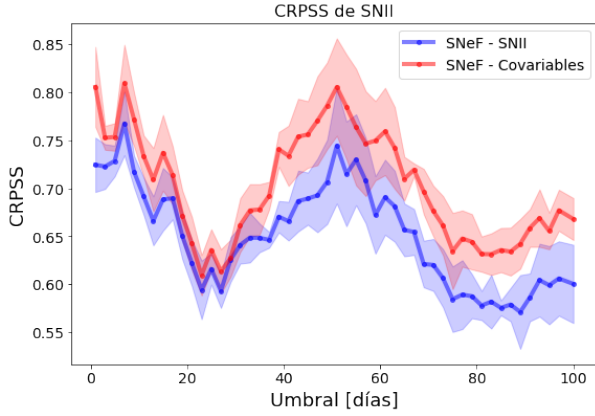
(d) sMAPE



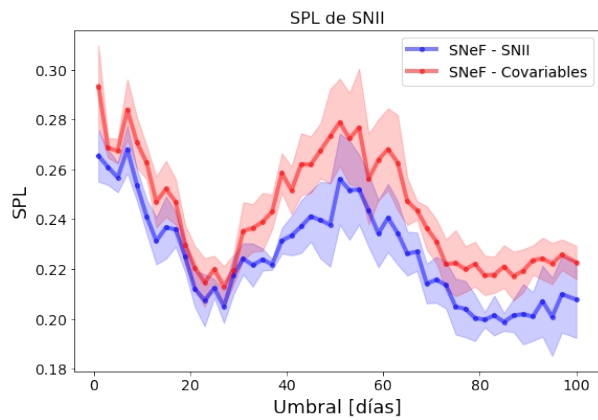
(b) PICP



(e) MSIS

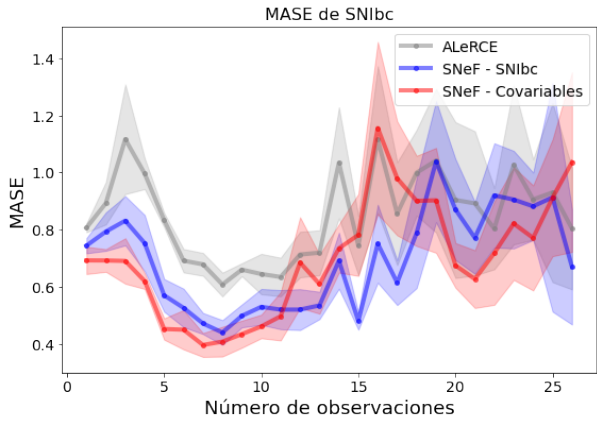


(c) CRPSS

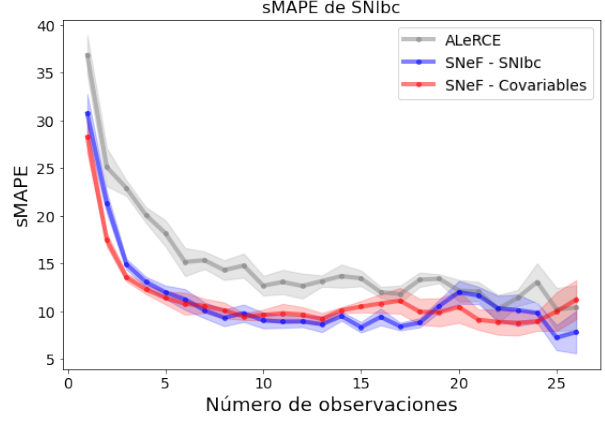


(f) SPL

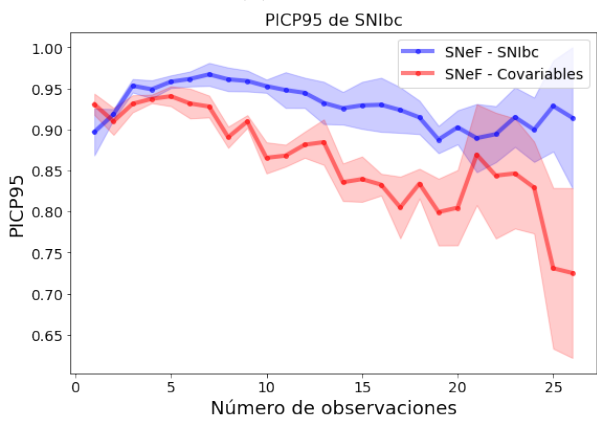
Figura A.2: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALerCE-P, **SNeF-SNII** (modelo entrenado solo con supernovas de tipo SNII) y **SNeF - Covariables** (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



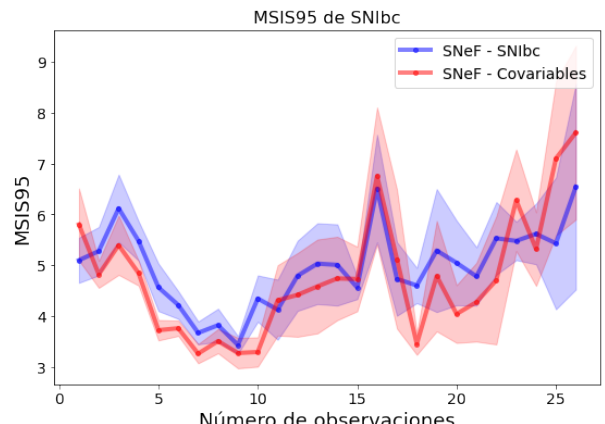
(a) MASE



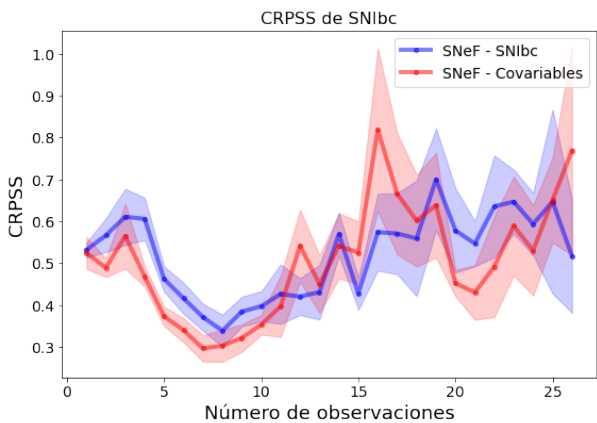
(d) sMAPE



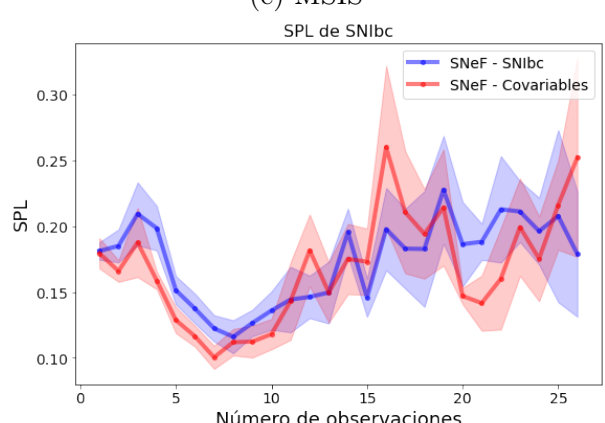
(b) PICP



(e) MSIS



(c) CRPSS



(f) SPL

Figura A.3: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALerCE-P, **SNeF-SNIbc** (modelo entrenado solo con supernovas de tipo SNIbc) y **SNeF - Covariables** (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



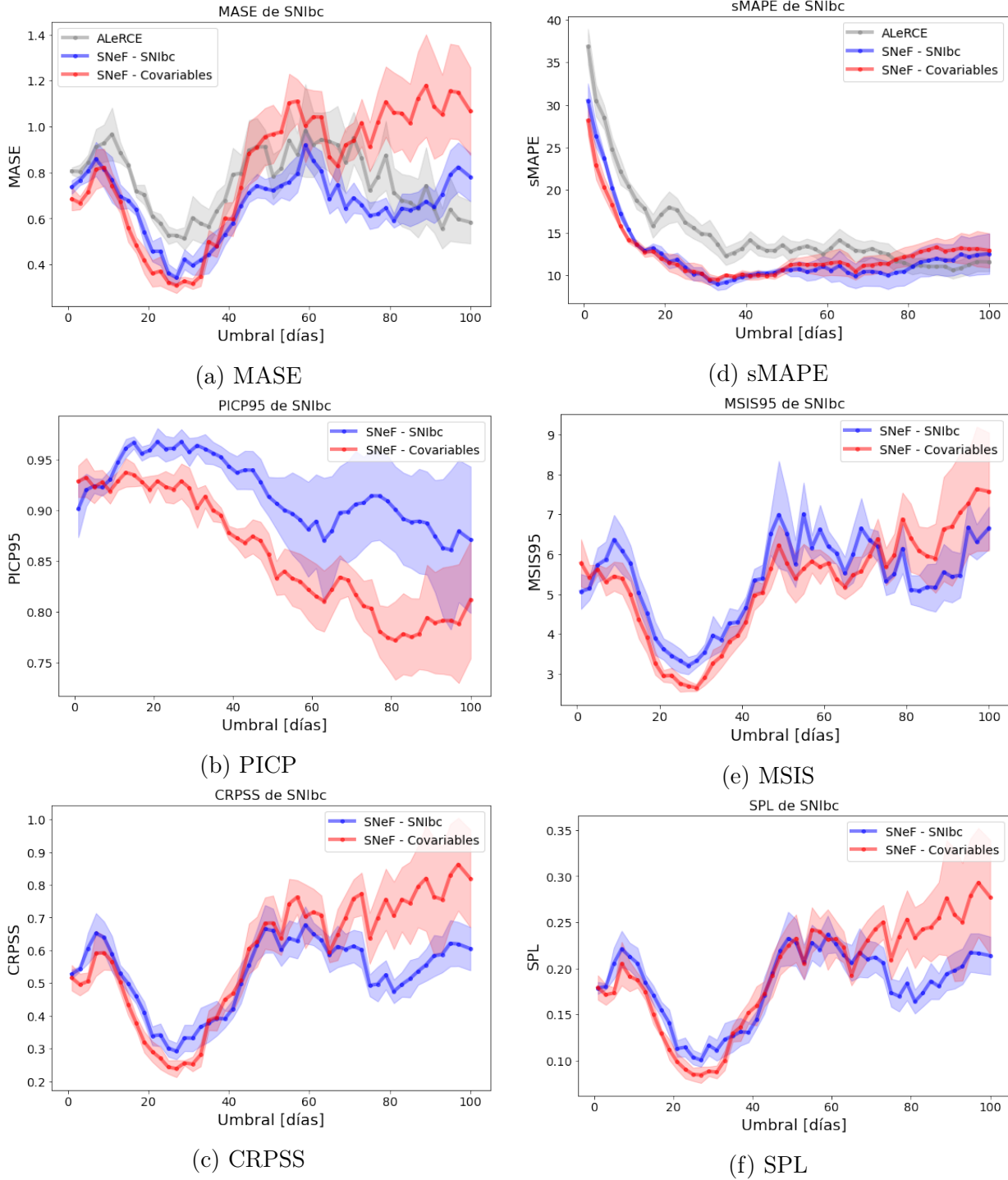
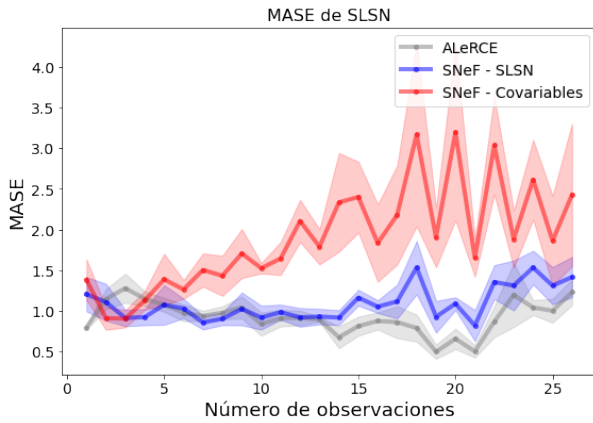
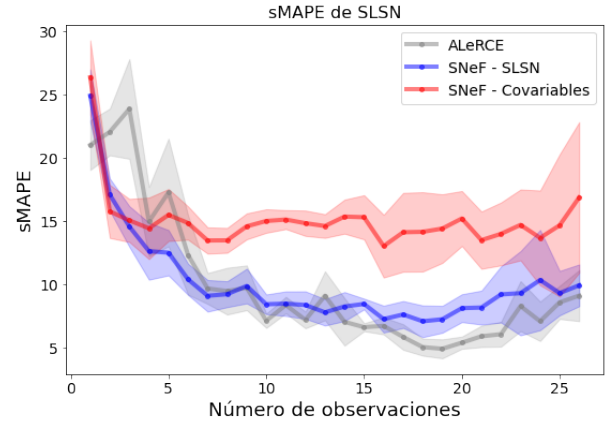


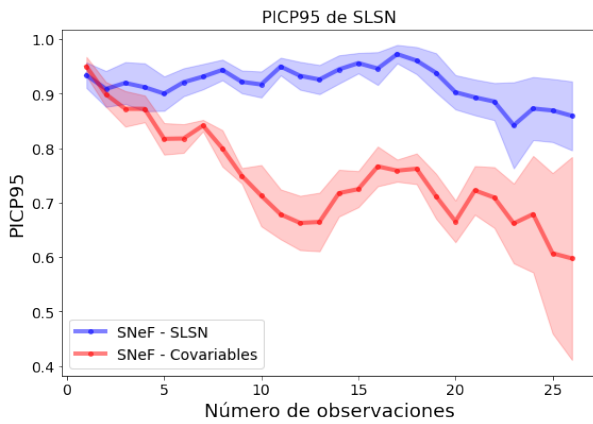
Figura A.4: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALeRCE-P, **SNeF-SNIbc** (modelo entrenado solo con supernovas de tipo SNIbc) y **SNeF - Covariables** (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



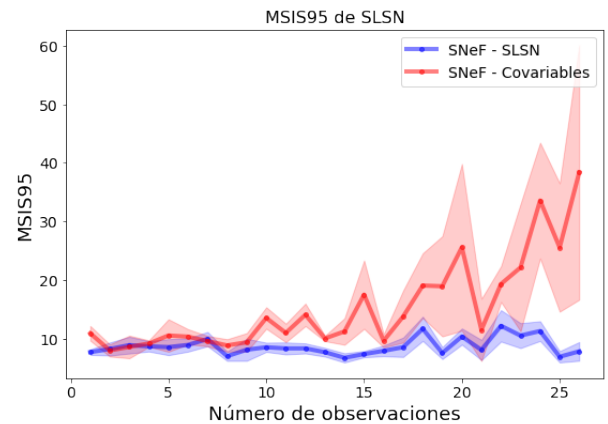
(a) MASE



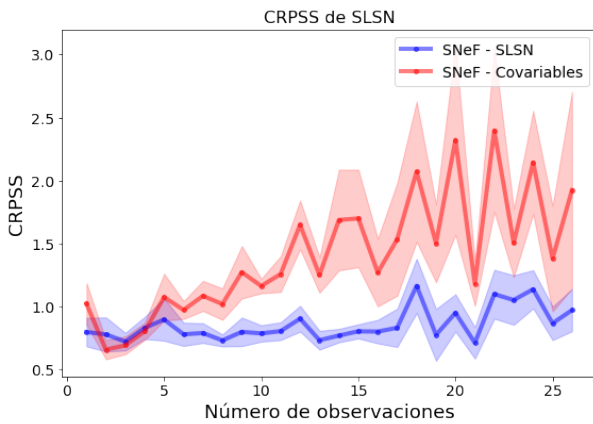
(d) sMAPE



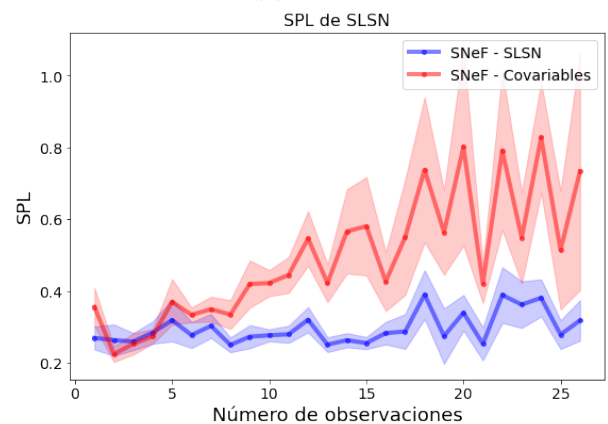
(b) PICP



(e) MSIS



(c) CRPSS



(f) SPL

Figura A.5: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALeRCE-P, **SNeF-SLSN** (modelo entrenado solo con supernovas de tipo SLSN) y **SNeF - Covariables** (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde a la mitad de la desviación estándar para simplificar la figura.

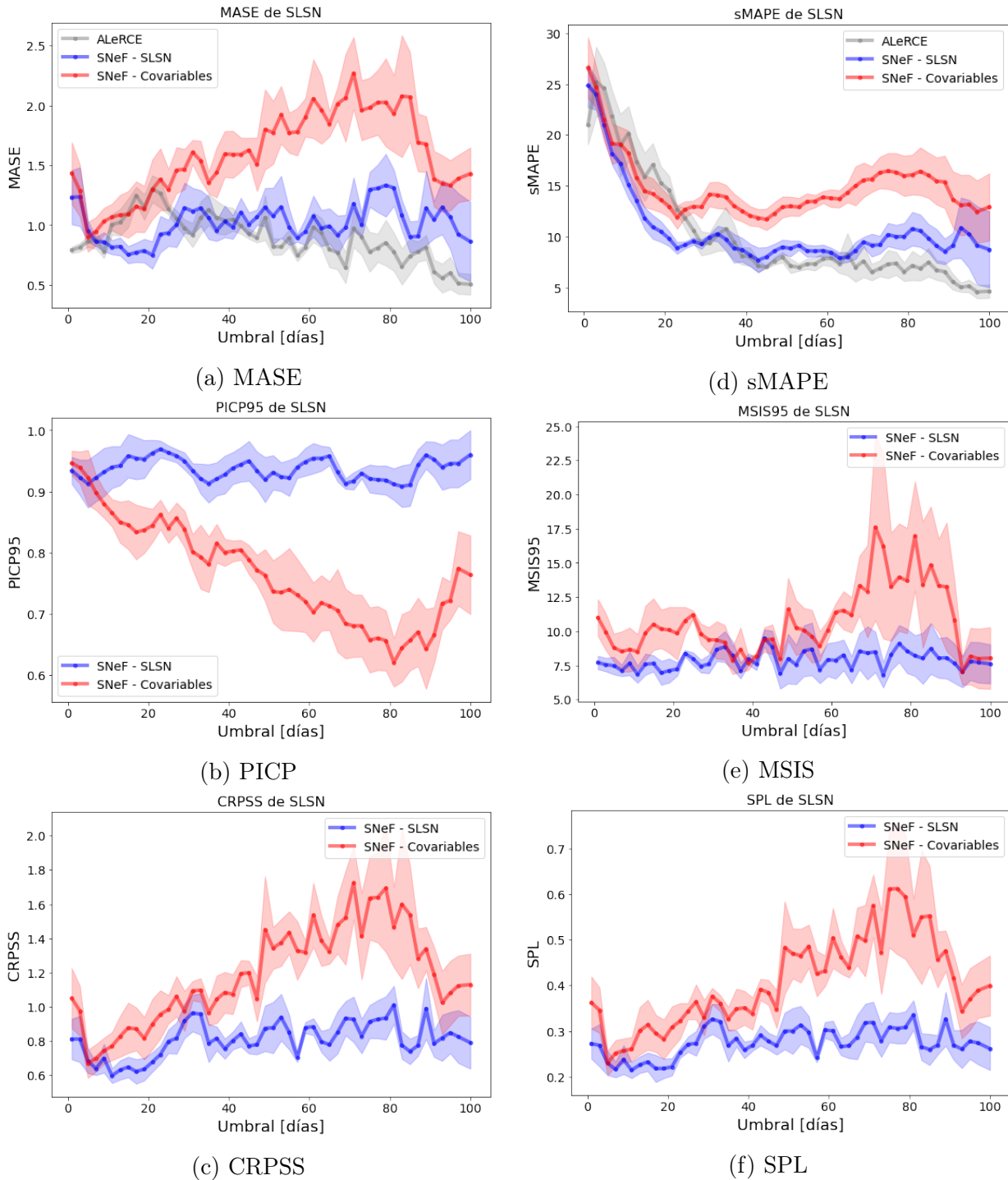
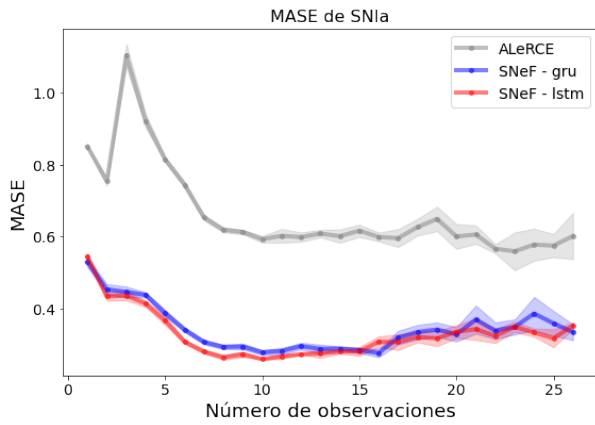


Figura A.6: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALeRCE-P, **SNeF-SLSN** (modelo entrenado solo con supernovas de tipo SLSN) y **SNeF - Covariables** (modelo entrenado con distintos tipos de supernovas). Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde a la mitad de la desviación estándar para simplificar la figura.

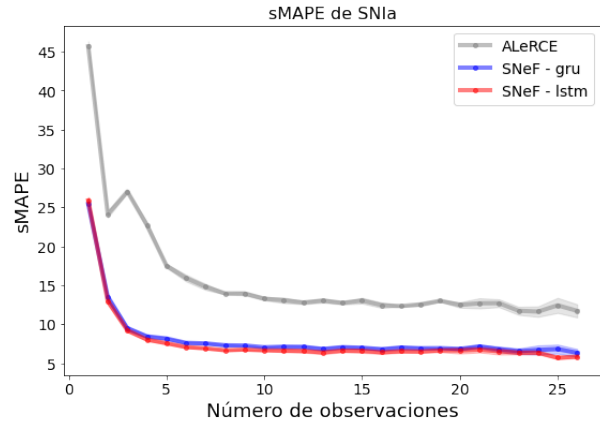
# Anexo B

## Métricas según celda recurrente

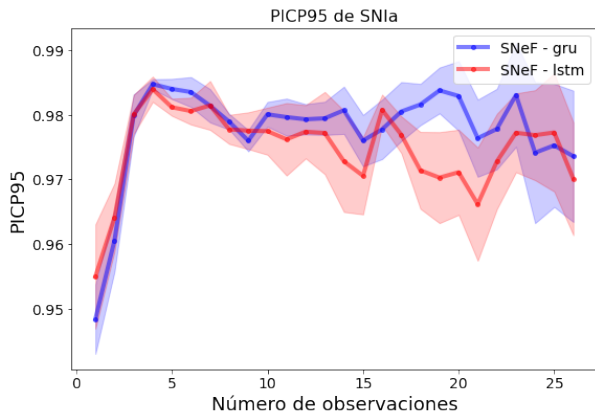
A continuación se muestra una comparación al usar el modelo SNeF - Covariables con distintas celdas recurrentes: GRU y LSTM. Esto se realiza para cada una de los tipos de supernovas.



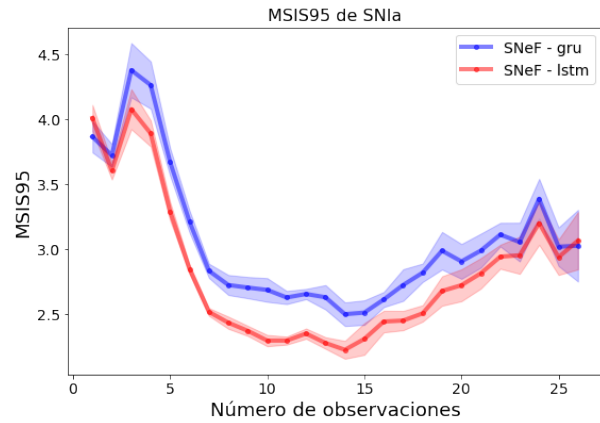
(a) MASE



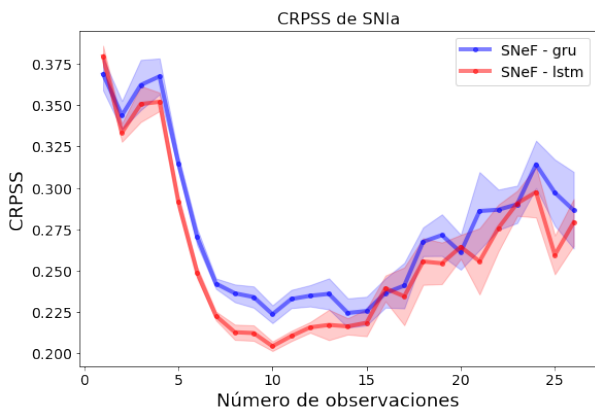
(d) sMAPE



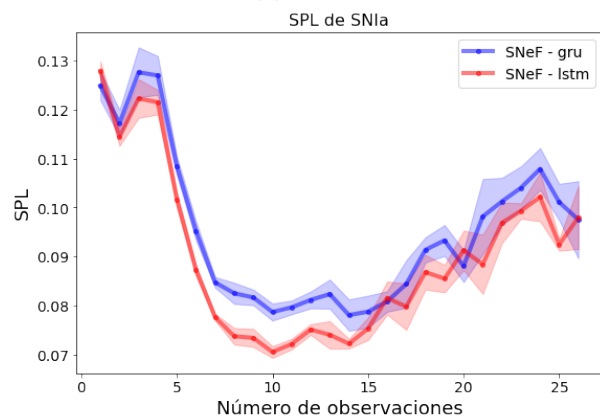
(b) PICP



(e) MSIS

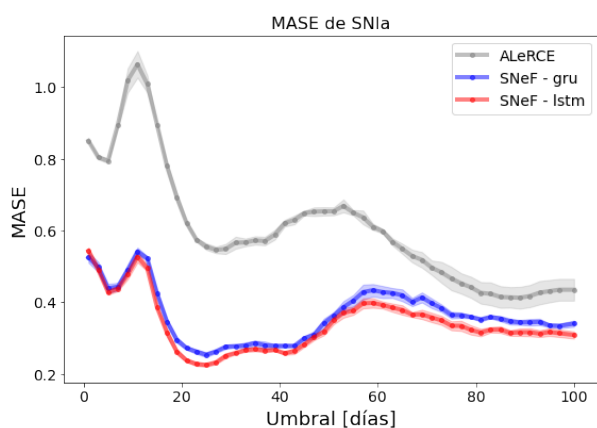


(c) CRPSS

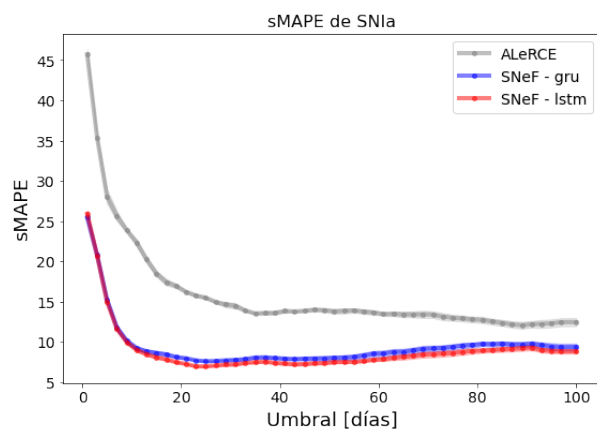


(f) SPL

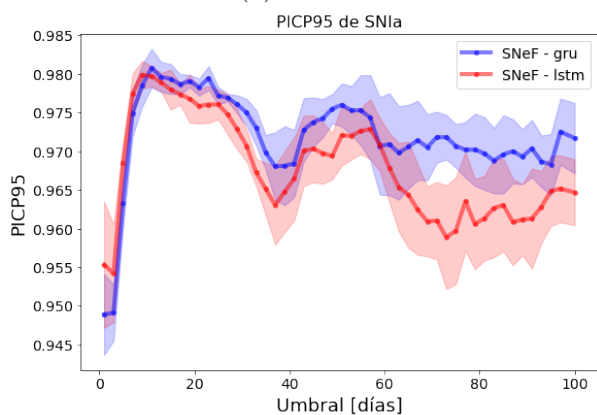
Figura B.1: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIa. Se comparan los resultados para el modelo ALerCE-P, **SNeF - gru** y **SNeF - lstm**. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



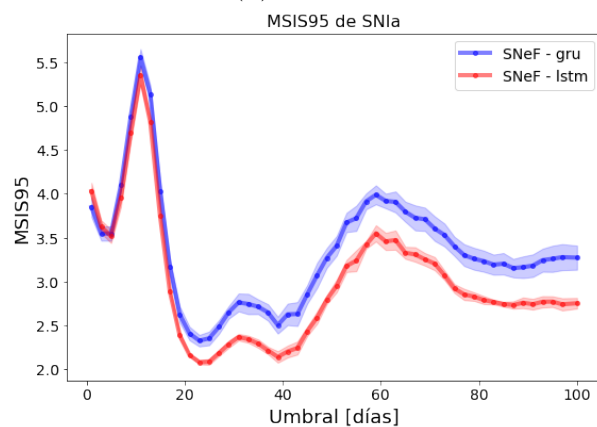
(a) MASE



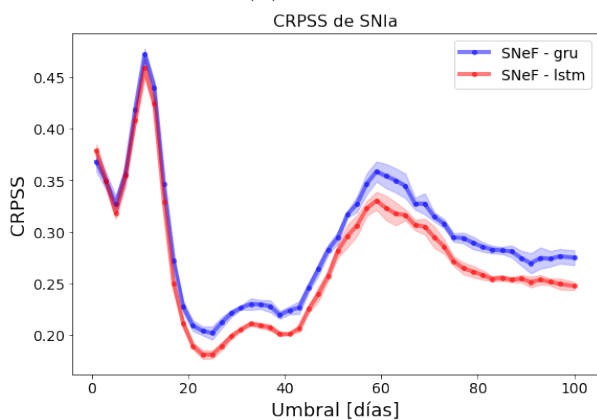
(d) sMAPE



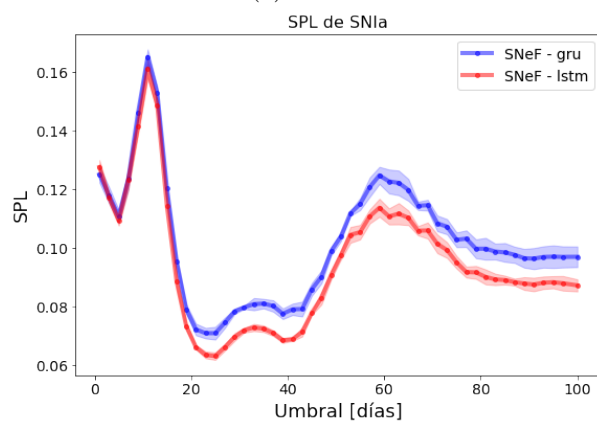
(b) PICP



(e) MSIS

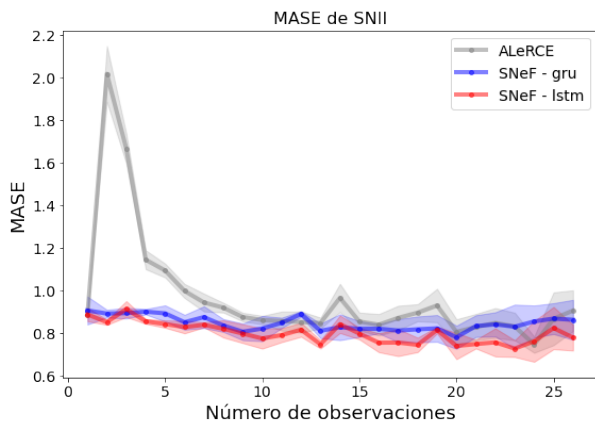


(c) CRPSS

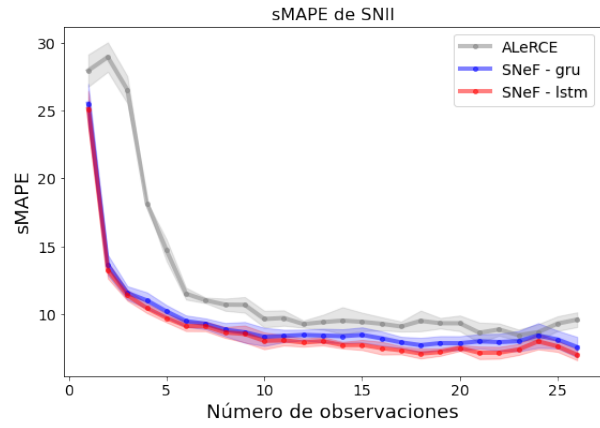


(f) SPL

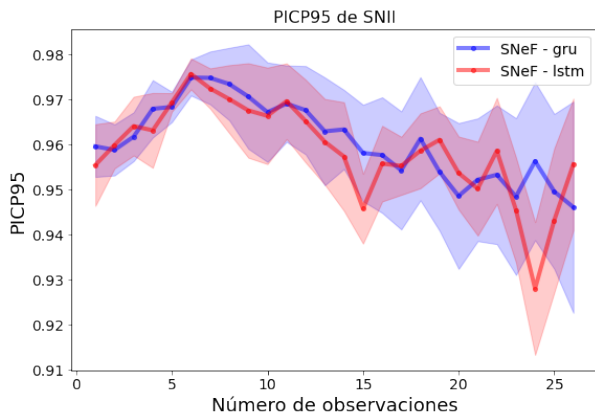
Figura B.2: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNIa. Se comparan los resultados para el modelo ALerCE-P, **SNeF - gru** y **SNeF - lstm**. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



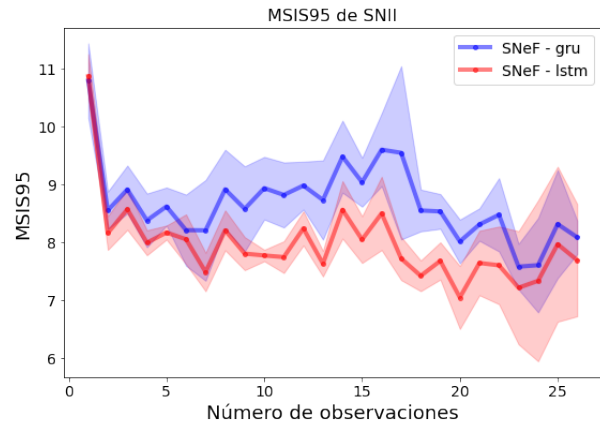
(a) MASE



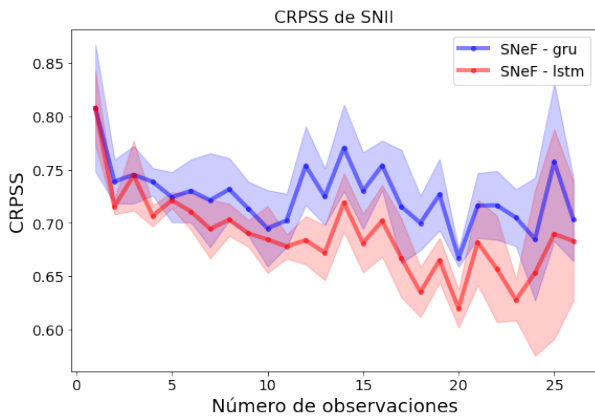
(d) sMAPE



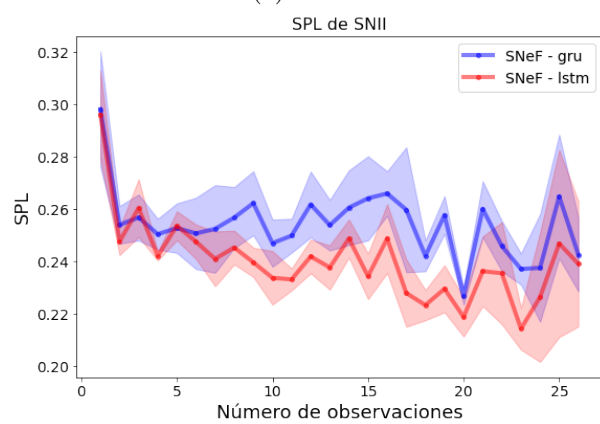
(b) PICP



(e) MSIS

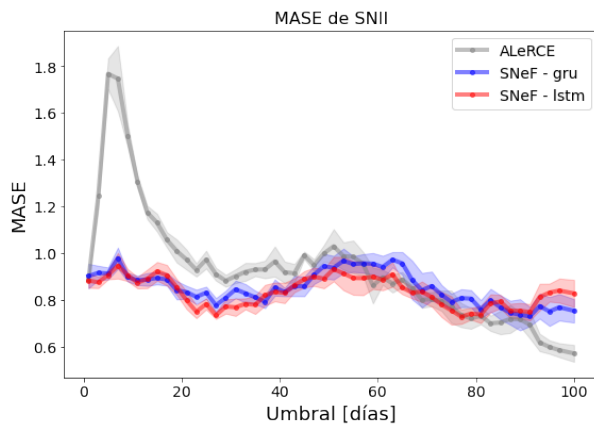


(c) CRPSS

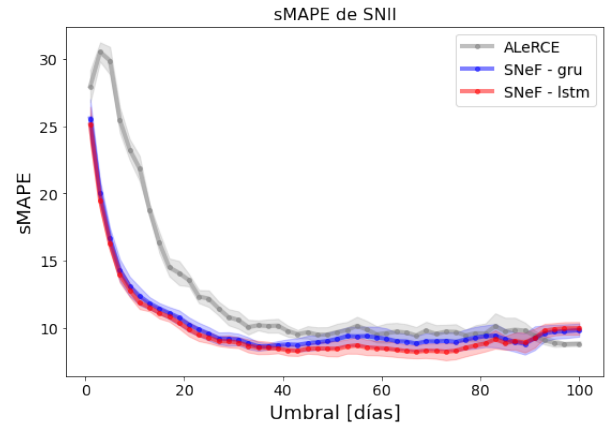


(f) SPL

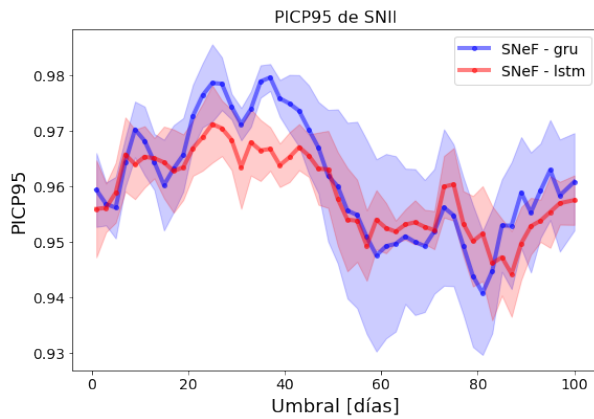
Figura B.3: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIi. Se comparan los resultados para el modelo ALerCE-P, **SNeF - gru** y **SNeF - lstm**. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



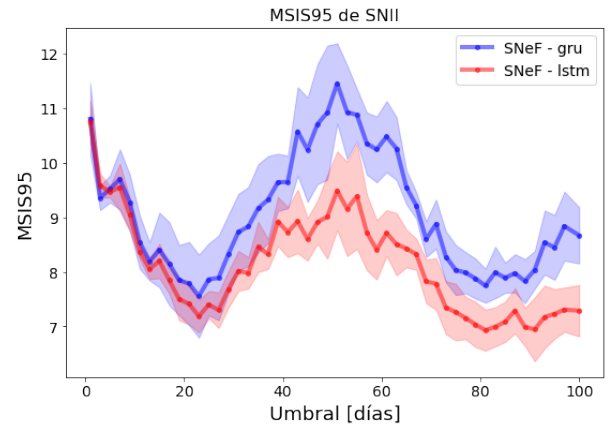
(a) MASE



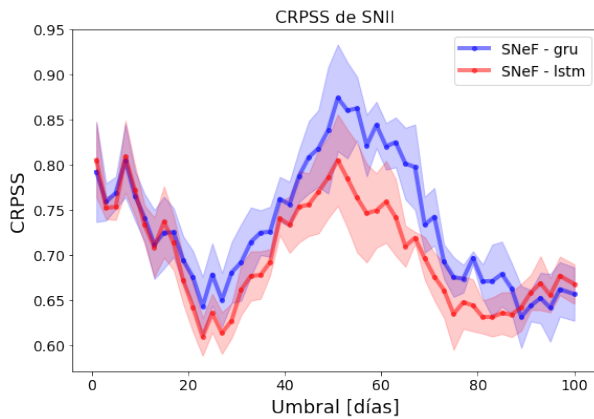
(d) sMAPE



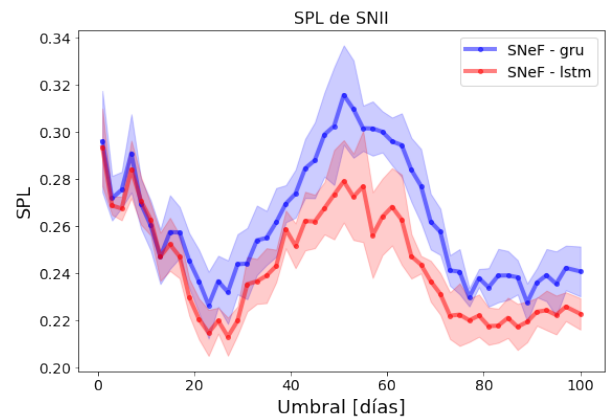
(b) PICP



(e) MSIS



(c) CRPSS



(f) SPL

Figura B.4: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNII. Se comparan los resultados para el modelo ALerCE-P, **SNeF - gru** y **SNeF - lstm**. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



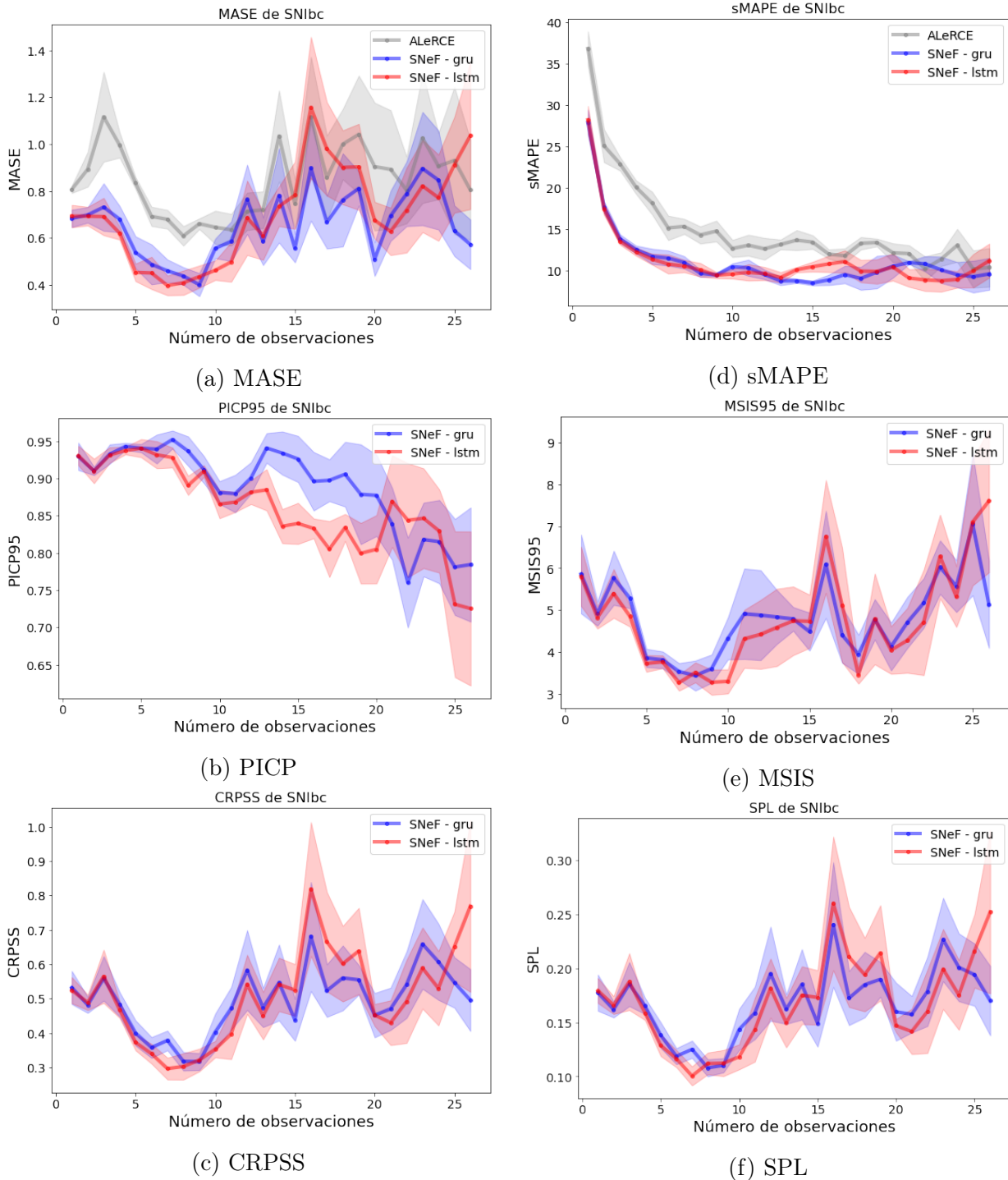
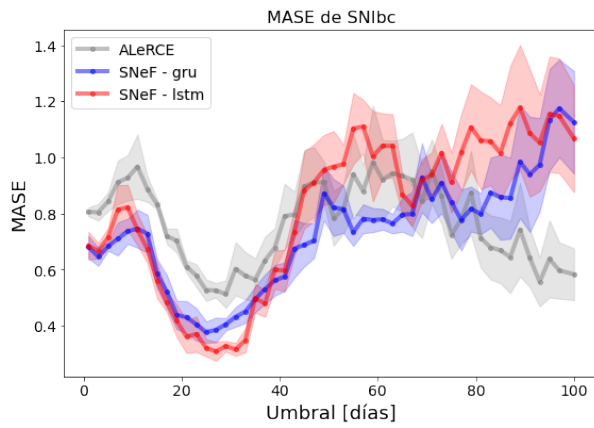
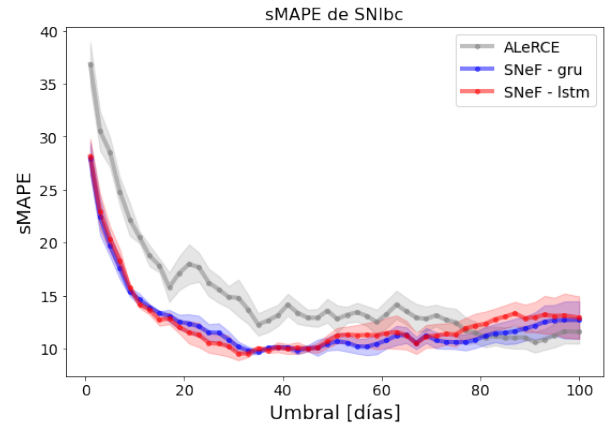


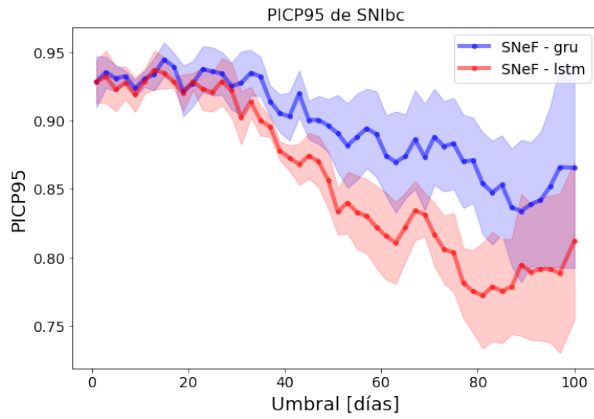
Figura B.5: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALerCE-P, SNeF - gru y SNeF - lstm. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



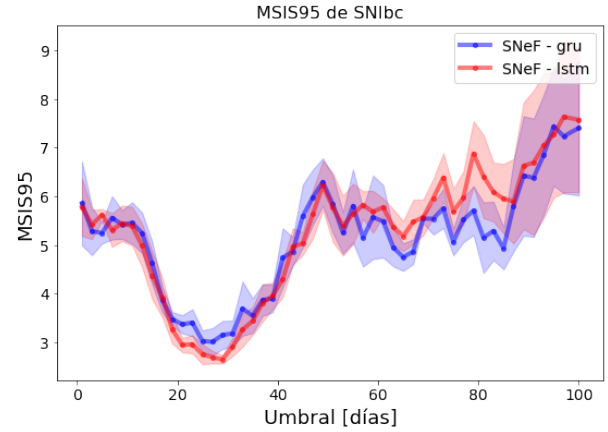
(a) MASE



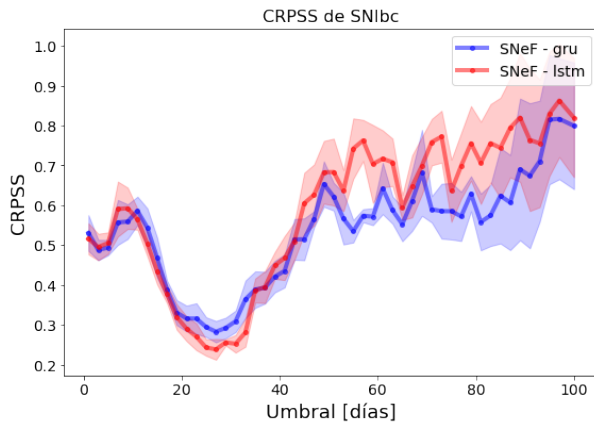
(d) sMAPE



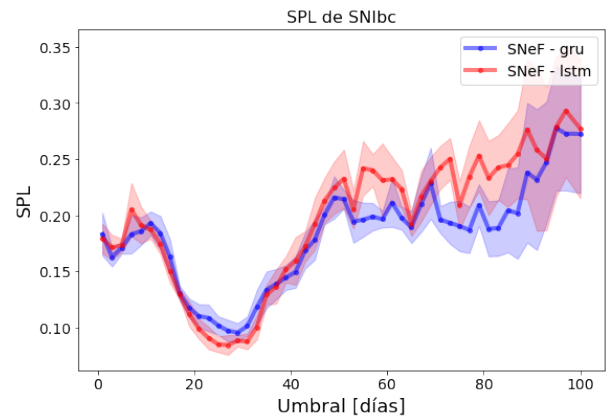
(b) PICP



(e) MSIS



(c) CRPSS



(f) SPL

Figura B.6: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SNIbc. Se comparan los resultados para el modelo ALerCE-P, **SNeF - gru** y **SNeF - lstm**. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.

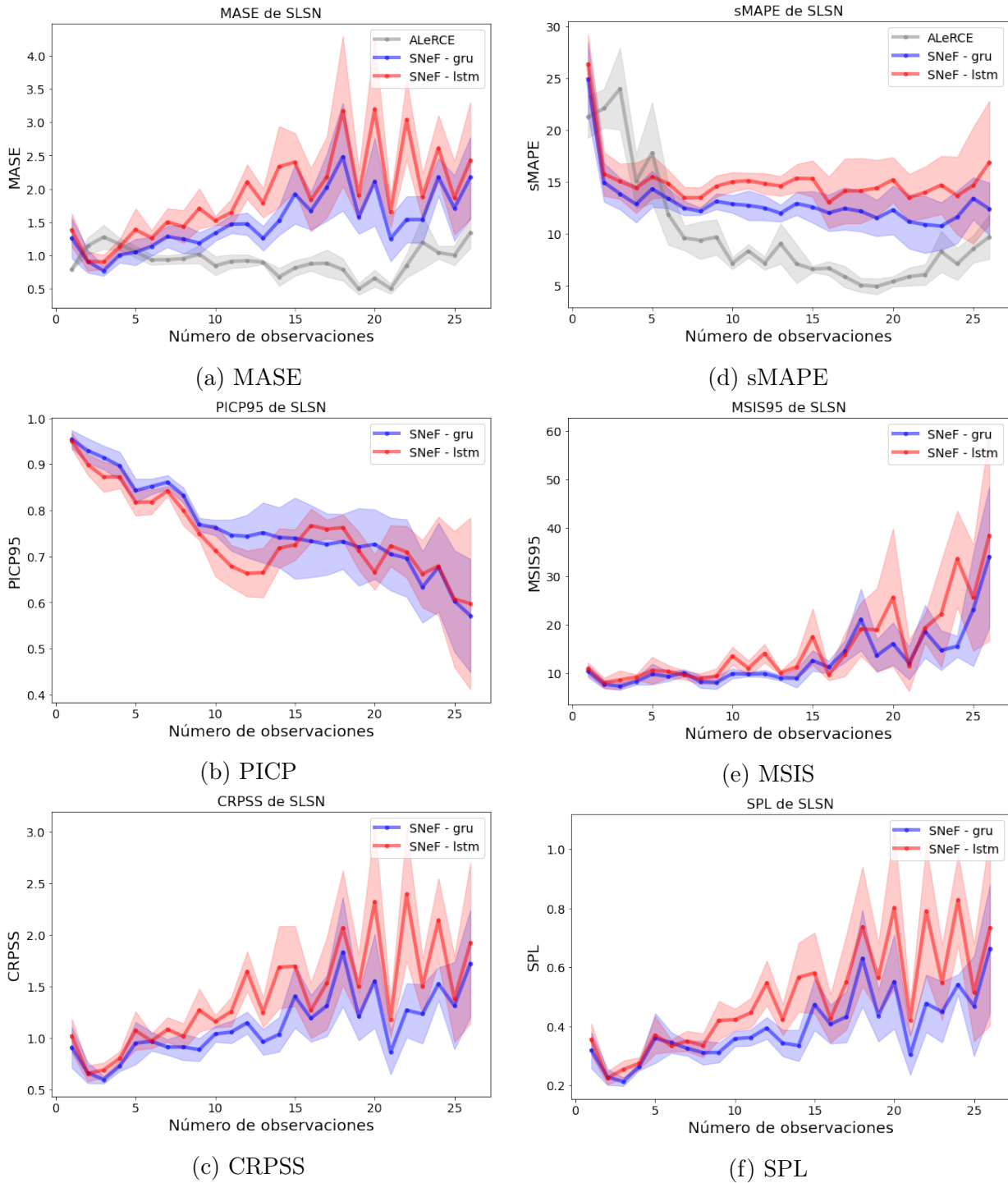
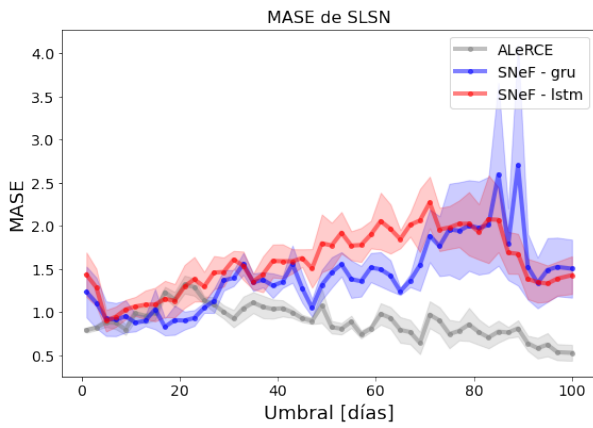
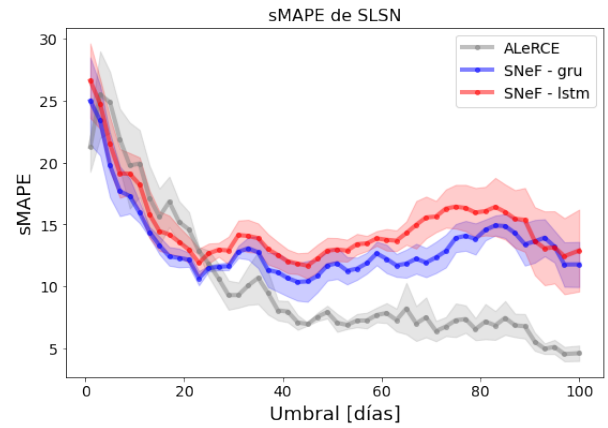


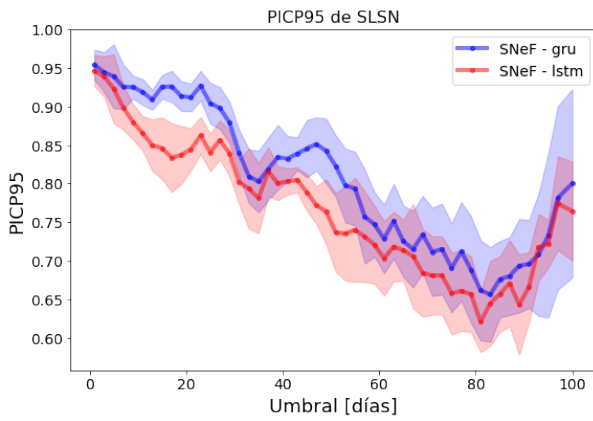
Figura B.7: Conjunto de métricas obtenidas en función del número de observaciones, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALerCE-P, SNeF - gru y SNeF - lstm. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.



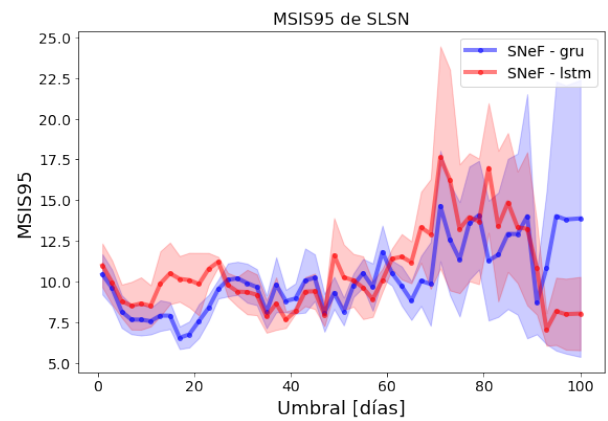
(a) MASE



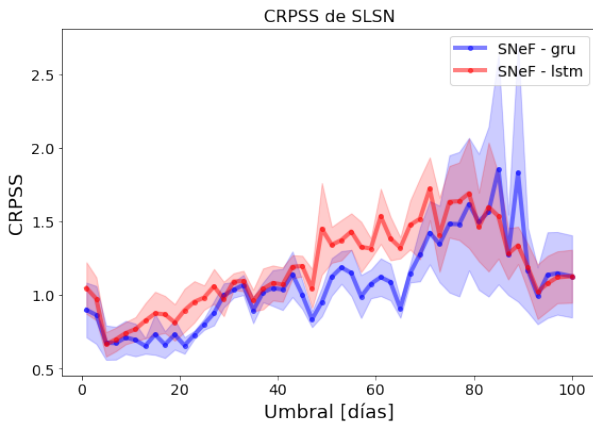
(d) sMAPE



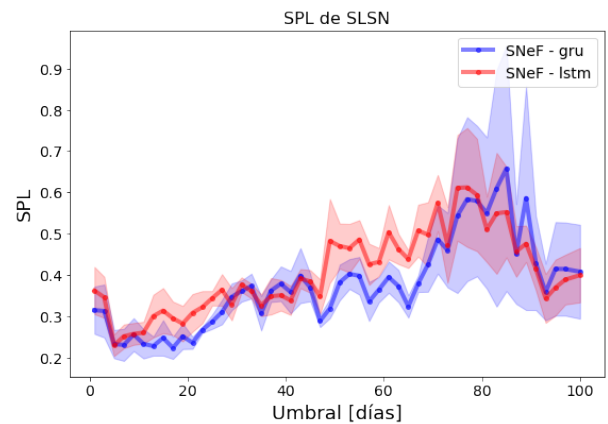
(b) PICP



(e) MSIS



(c) CRPSS



(f) SPL

Figura B.8: Conjunto de métricas obtenidas en función del umbral de días, evaluadas en el conjunto de supernovas de tipo SLSN. Se comparan los resultados para el modelo ALerCE-P, **SNeF - gru** y **SNeF - lstm**. Los resultados corresponden al promedio de las métricas de 5 particiones distintas de los datos. El área sombreada corresponde la mitad de la desviación estándar para simplificar la figura.

# Anexo C

## Ejemplos de predicciones

En esta sección se presentan ejemplos de predicciones para los distintos tipos de supernovas (Ver Figuras C.1, C.2, C.3, C.4). El modelo utilizado corresponde a SNeF - Covariables, el cuál fue entrenado con todos los tipos de supernovas.

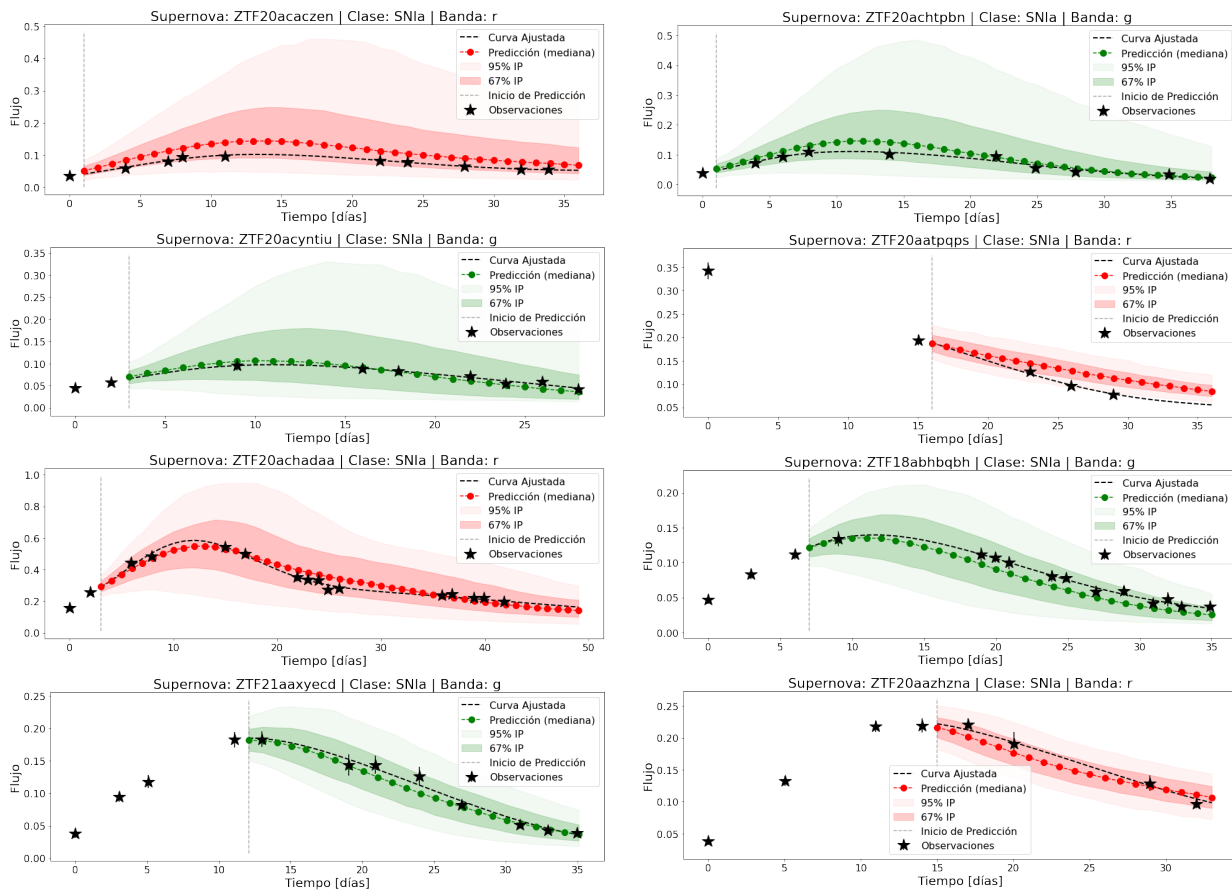


Figura C.1: Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SNIa. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales.

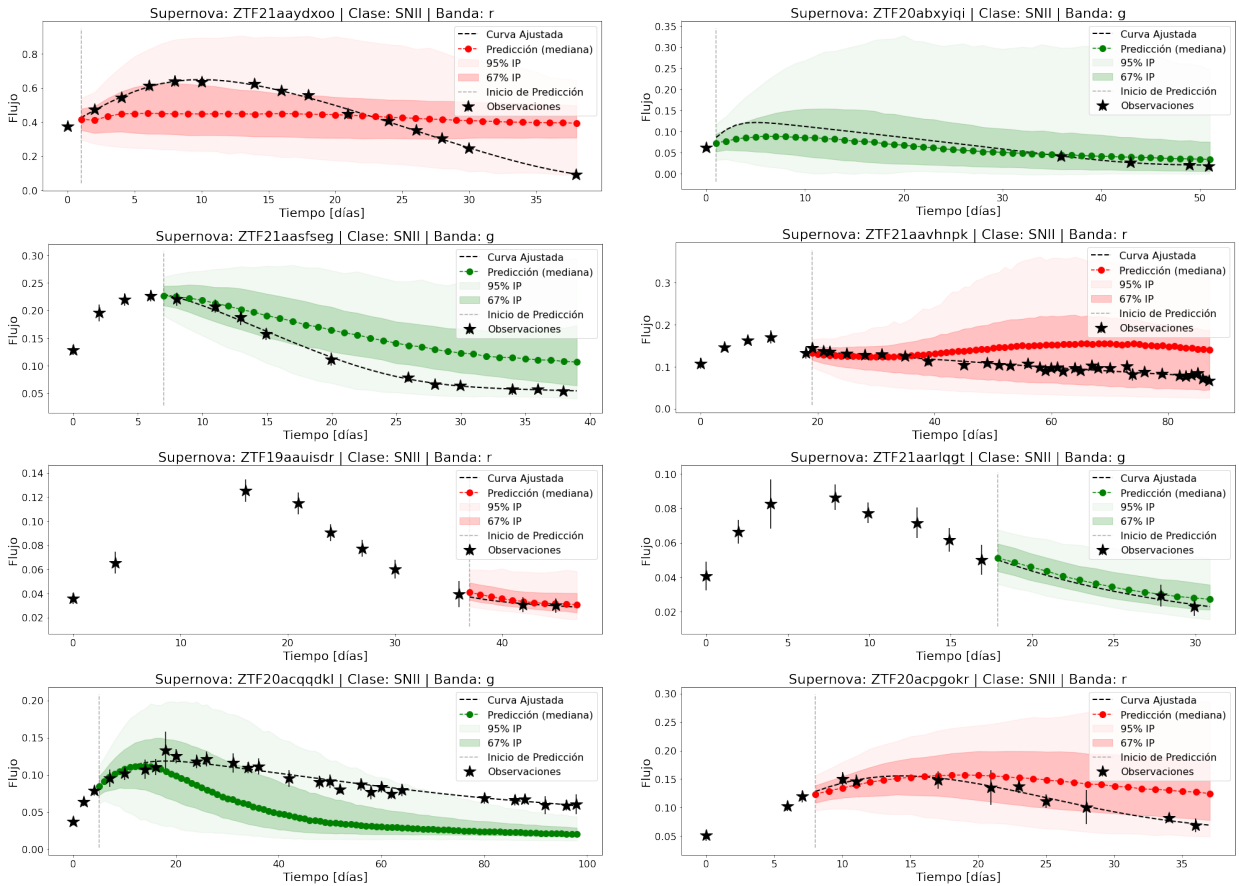


Figura C.2: Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SNII. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales.

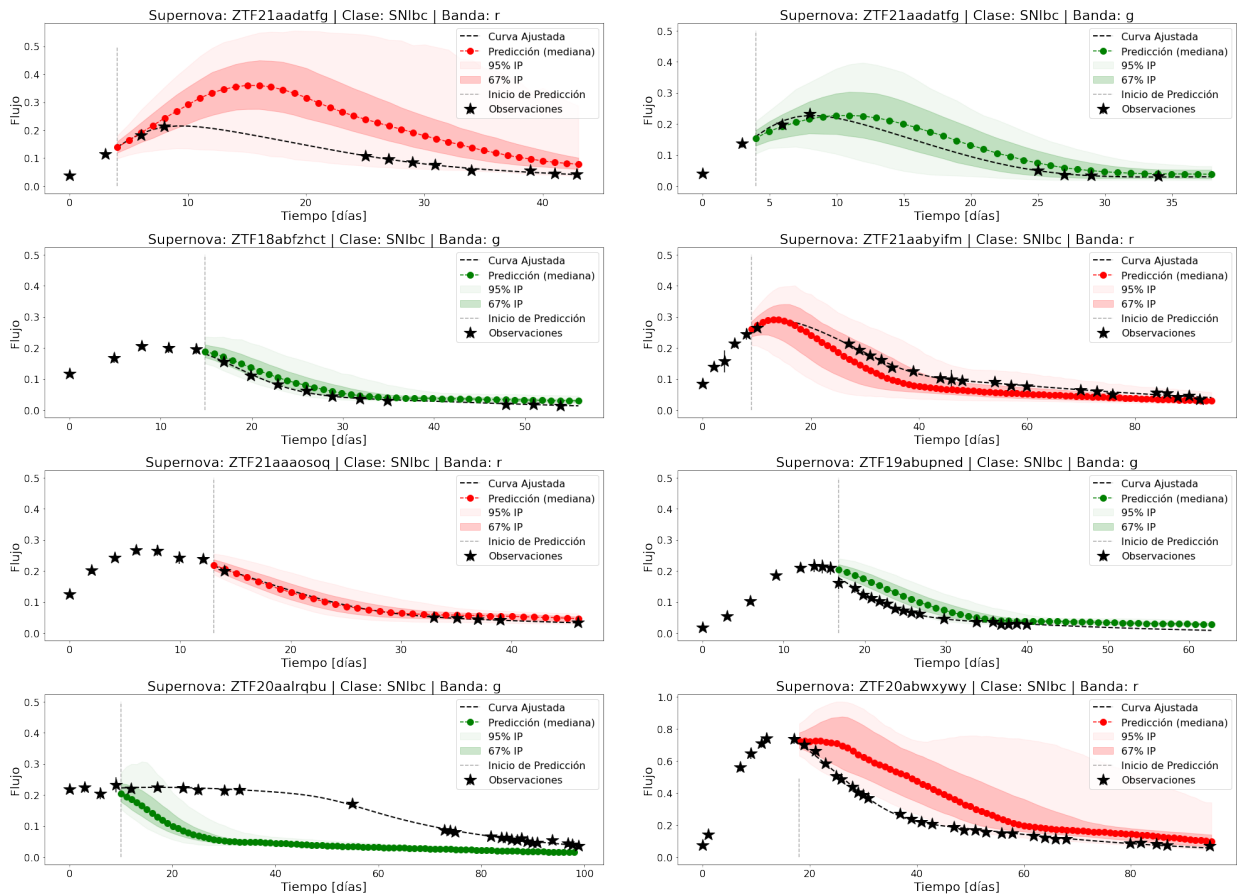


Figura C.3: Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SNIbc. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales.

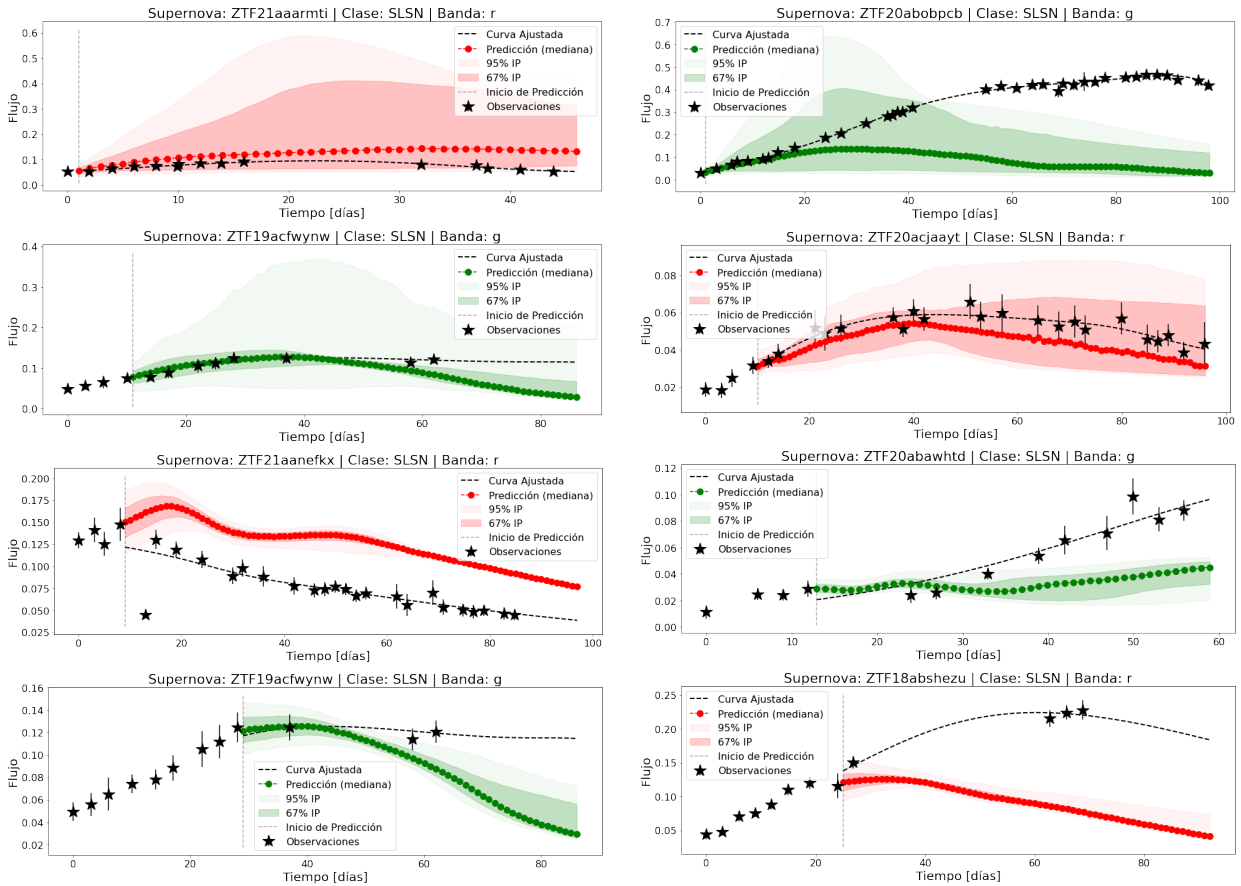


Figura C.4: Ejemplos de predicciones de 8 curvas de luz de supernovas tipo SLSN. En cada uno de los gráficos se observa la comparación entre la curva ajustada y al modelo de predicción propuesto: SNeF - Covariables. Es importante notar que los datos entrantes corresponden solo a los datos que se encuentren a la izquierda de la línea vertical punteada (inicio de predicción), donde el resto de observaciones son referenciales.