Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Programa de Maestría en Computación

Propuesta y evaluación de una estrategia para imputación múltiple y multivariada de valores faltantes en series de tiempo del campo meteorológico utilizando aprendizaje automático

(Proposal and evaluation of a strategy for multiple and multivariate imputation of missing values in time series of the meteorological field using machine learning)

Tesis para optar por el grado de *Magister Scientiae* en Computación

Autor:
Ana Cristina Arias Muñoz

Profesor Asesor:
Luis Alexánder Calvo Valverde

Junio, 2022

Resumen

Muchas de las decisiones importantes dependen de los resultados de las predicciones: evaluar productividad futura de un producto, prestamistas que buscan estimar la recuperación y ganancias sobre el capital, diagnósticos médicos, prónosticos del clima con diferentes fines (agricultura, turismo, meteorología, etcétera), evaluación de personal para decidir contrataciones en empresas, analizar postulantes para otorgar admisiones y becas en universidades, entre otros [1] [2].

Los pronósticos se basan en el análisis de series de tiempo para entender comportamientos, estacionalidades y tendencias en los datos. Sin embargo, es común que los datos registrados puedan tener faltantes a la hora de recopilar dicha información. La razones pueden ser variadas: pérdida de conectividad a internet, utilización de energía solar en los sensores (que pueden dañarse o no tener suficiente carga), corto circuitos, sensores húmedos o dañados, problemas de carga o estado en baterías, etcétera.

A pesar de que actualmente ya existen estudios y algoritmos utilizados en la imputación de datos faltantes en series de tiempo, es de interés de esta investigación evaluar y proponer una arquitectura que utilice aprendizaje automático para la imputación de datos en series de tiempo utilizando múltiples variables y varias estaciones climatológicas de una misma zona geográfica para lograr la imputación de datos y comparar su efectividad con respecto algunas metologías conocidas en la imputación de series de tiempo.

Los resultados de esta investigación señalan que MissForest y EMB siguen siendo algoritmos robustos para la imputación de datos faltantes, sin embargo la arquitectura de aprendizaje automático propuesta es capaz de imputar al 1 %, 3 % y 5 % de datos faltantes con métricas que compiten con los algoritmos supra mencionados con la ventaja que se pueden imputar los datos faltantes de una estación a través del modelo entrenado con alguna otra estación cercana que comparta características similares a la estación que se pretende imputar. Sin embargo, para datos faltantes al 10 % y 20 % los algoritmos EMB y MissForest siguen siendo superiores a la arquitectura propuesta en esta investigación.

Abstract

Many important decisions are based on the outcomes of forecasts: estimating future productivities, forecasting capital recovery and profits, estimating medical diagnoses, forecasting climate for various purposes (agriculture, tourism, meteorology, etc.), evaluating employees to make hiring decisions, analyzing candidates for administrative and financial positions in unions, and so on[1] [2].

Prognostications are based on the analysis of time series to understand behaviors, trends, and patterns in data. However, it is common for registered data to be missing when attempting to reassemble this information. The reasons for this can vary: loss of internet connectivity, use of solar energy in sensors (which can cause damage or not have enough capacity), short circuits, damaged sensors, battery problems, etc.

Despite the fact that there are currently studies and algorithms used in the imputation of missing data in time series, it is the goal of this research to evaluate and propose an architecture that uses machine learning for data imputation in time series by utilizing multiple variables and various climatological stations in a single geographical area to achieve data imputation and prediction.

The findings of this study indicate that MissForest and EMB continue to be underutilized algorithms for the imputation of missing data. However, the proposed auto-learning architecture is capable of imputing up to 1 %, 3 %, and 5 % of missing data resulting in metrics that compete with the previously mentioned algorithms, with the added benefit of being able to impute data from multiple locations. However, for missing data between 10 % and 20 %, the EMB and MissForest algorithms continue to outperform the architecture proposed in this study.

Dedicatoria

A mi familia por todo el apoyo a través de los años de estudio, a mi compañero de vida por estar siempre a mi lado, ayudarme a creer en mí aún cuando he creído no poder más y a mis hijos por darle sentido a mi vida y ayudarme a buscar ser ejemplo de vida para ustedes, ! que el cielo sea nuestro límite!

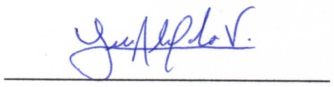
Agradecimientos

Agradezco a mi profesor asesor, Dr. Luis Alexánder Calvo, por todo el apoyo, tiempo dedicado a lo largo de todo este proyecto. A los lectores, por sus contribuciones valiosas. A Susana Cob García, por todo el apoyo, acompañamiento, trabajo y sugerencias buscando siempre mejorar el trabajo. A Vannesa Rojas de ICAFE por el acompañamiento y brindar información y herramientas para colaborar en esta investigación.

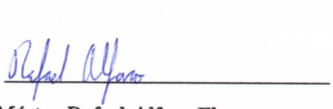
Aprobación de tesis

Propuesta y evaluación de una estrategia para imputación múltiple y multivariada de valores faltantes en series de tiempo del campo meteorológico utilizando aprendizaje automático.

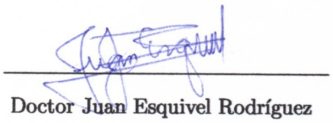
Tribunal examinador



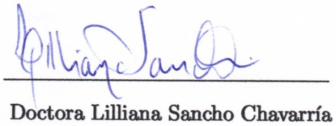
Doctor Luis Alexánder Calvo Valverde
Profesor asesor



Máster Rafael Alfaro Flores
Lector externo



Doctor Juan Esquivel Rodríguez
Lector



Doctora Lilliana Sancho Chavarría
Coordinadora del Programa de Maestría

Junio, 2022

Glosario

ICAFE: El Instituto del Café de Costa Rica (ICAFE) es una institución pública de carácter no estatal, fundada en 1933 como rectora de la caficultura costarricense, con el objetivo de promover un modelo de producción único y equitativo entre los productores, beneficios, tostadores y exportadores nacionales, dedicado a investigar y desarrollar tecnología agrícola e industrial.

Random forest: Modelo de clasificación basado en árboles de decisión aleatorios.

Overfitting: Término en inglés para referirse a un modelo de clasificación que aprende de patrones de los datos de entrenamiento que no son generalizables a los datos de prueba, generando que el rendimiento del modelo sea significativamente menor durante la prueba que durante entrenamiento.

Bootstrapping: Término en inglés para aproximar el sesgo o la varianza de un análisis estadístico, así como para construir intervalos de confianza.

Imputación: En estadística, la imputación es la sustitución de valores no informados en una observación por algún valor a partir de otros valores válidos en otras observaciones.

Jupyter Notebook: Es un proyecto y una comunidad donde el objetivo es desarrollar software libre, con estándares abiertos y servicios para una computación interactiva sobre docenas de lenguajes de programación.

Google Colab: Es un cuaderno Jupyter en línea que permite escribir y compartir cuadernos Jupyter con quien quiera através de la cuenta de Google. Es equivalente a un ambiente local de trabajo de programación, pero en la nube, por lo que no requiere instalación de software y está disponible solamente con una conexión a internet.

Wandb: Es una plataforma de aprendizaje automático basada en seguimiento de experimentos, versionamiento de conjunto de datos y evaluaciones de modelos.

Keras Tuner: Es una herramienta muy sencilla de utilizar que simplifica en gran medida la complejidad de aplicar un proceso de hiperparametrización sobre redes neuronales profundas construidas mediante Keras, ofreciendo un amplio grado de versatilidad para optimizar tanto la estructura de la red como la configuración de los parámetros.

Tabla de contenido

1. Introducción **17**

2. Marco Teórico **19**

3. Descripción de la Investigación **33**

3.1. Trabajos relacionados: "Datos faltantes en series de tiempo"(estado del arte) 33

3.2. Objetivos 44

3.2.1. Objetivo general 44

3.2.2. Objetivos específicos 44

3.3. Hipótesis 44

3.4. Métricas 45

4. Diseño Experimental **46**

4.1. Descripción de experimentos 46

4.1.1. Descripción del Problema 46

4.1.2. Variables de Respuesta 46

4.1.3. Factores 47

4.1.4. Selección del diseño experimental 48

4.1.5. Análisis estadístico 49

4.2. Detalles de los experimentos 49

4.2.1. Ambiente de desarrollo 49

4.2.2. Conjuntos de datos 50

4.2.3. Preprocesamiento de datos 53
4.2.4. Generación de datos faltantes para evaluación de imputaciones 66

5. Arquitectura propuesta de aprendizaje automático para la imputación múltiple y multivariada de valores faltantes en series de tiempo del campo meteorológico utilizando LSTM 68

5.0.1. Arquitectura propuesta de aprendizaje automático para la imputación múltiple y multivariada de valores faltantes en series de tiempo del campo meteorológico utilizando LSTM con conjuntos de datos correlacionados por variable según las estaciones aledañas 72
5.0.2. Hiperparametrización 74
Hiperparametrización del modelo EMB 75
Hiperparametrización del modelo Random Forest: Missforest . 75
Hiperparametrización del modelo LSTM propuesto 75

6. Experimentos finales y análisis de resultados 77

7. Conclusiones y Trabajo futuro 99

7.1. Conclusiones 99
7.2. Trabajo futuro 101

Bibliografía 109

Índice de figuras

2.1. Representación de una neuronal LSTM: tomado de [3] 27
2.2. Encoder-Decoder [4] 28
2.3. Imagen ejemplo de Random Forest [5] 30
2.4. Esquema del algoritmo EMB. Fuente: tomado de [6] 31

3.1. Comparación de los efectos entre los algoritmos de DA y EMB. Fuente: tomado de [7] 35
3.2. Efectos de la imputación de datos en seis tasas faltantes. Fuente: tomado de [7] 35
3.3. Porcentaje de observaciones faltantes en conjunto de datos de velocidad de tráfico en reconocimiento de patrones de letras. 37
3.4. Porcentaje de observaciones faltantes en patrones de flujo de agua en series de tiempo. Fuente: Tomado de [8] 37
3.5. Porcentaje de observaciones faltantes en patrones del conjunto de datos Nottem. Fuente: Tomado de [8] 37
3.6. Distribución de la lluvia y concetraciones de nitrato durante Junio 2017 y Mayo 2018. Fuente: Tomado de [9] 39
3.7. Evaluación de datos imputados para las métricas RMSE y MAE en 3 casos. Fuente: Tomado de [9] 40
3.8. Gráfico Box-whisker para el MAE derivado de SSIM en los tres casos experimentales. Fuente: Tomado de [9] 40
3.9. Ejemplo de alineamiento propio. Fuente: Tomado de [10] 41

3.10. Ejemplo de valor omitido. Fuente: Tomado de [10] 42
3.11. Comparación de rendimiento de las tareas de imputación. 43

4.1. Presencia de datos en las 32 estaciones de ICAFE 52
4.2. Archivos de intervalos generados de la estación 9, se muestra el rango de tiempo de datos faltantes 55
4.3. Filas: Identificación de la estación, columnas: año, y en cada celda la presencia de los datos originales en intervalos de cada día, facilitados por ICAFE 56
4.4. Ubicación geográfica de las estaciones metereológicas de la Zona de los Santos, Costa Rica. 56
4.5. Muestra diferentes subconjuntos de datos representados por diferentes colores 57
4.6. Temperatura Exterior en las estaciones 9, 10 11 y 12. 58
4.7. Humedad exterior en las estaciones 9, 10 11 y 12. 58
4.8. Temperatura máxima en las estaciones 9, 10 11 y 12. 59
4.9. Punto de rocío en las estaciones 9, 10 11 y 12. 59
4.10. Precipitación en las estaciones 9, 10 11 y 12. 59
4.11. Variables no correlacionadas en las estaciones 9, 10 11 y 12. 60
4.12. Variables en todas las estaciones 9, 10 11 y 12. 61
4.13. Distribución de los datos en la variable temperatura máxima estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal. 62
4.14. Distribución de los datos en la variable temperatura exterior estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal. 63
4.15. Distribución de los datos en la variable humedad exterior estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal. 63

4.16. Distribución de los datos en la variable punto de rocío estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal. 64
4.17. Distribución de los datos en la variable precipitación estación 9. Kurtosis y Skew no cumplen las afirmaciones para conciderarse una distribución normal. 65

5.1. Arquitectura LSTM propuesta 71
5.2. Construcción de nuevo conjunto de datos basado en la correlación entre las variables con las estaciones aledañas 73
5.3. Matriz de correlación entre estaciones y variables 74

6.1. Métricas obtenidas en la comparación de los algoritmos MissForest, EMB y el LSTM propuesto, imputando con los diferentes porcentajes valores faltantes. 82
6.2. Gráfica que analiza el RMSE de los algoritmos MissForest, EMB y LSTM propuesto en los diferentes porcentajes de datos faltantes . . . 82
6.3. Gráfica que analiza el MAE de los algoritmos MissForest, EMB y LSTM propuesto en los diferentes porcentajes de datos faltantes . . . 83
6.4. Gráfica que analiza el R2 de los algoritmos MissForest, EMB y LSTM propuesto en los diferentes porcentajes de datos faltantes 83
6.5. Experimento E. Imputación por variable Rain al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la precipitación es en milímetros cúbicos 91
6.6. Experimento E. Imputación por variable Dew Pt. al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida del punto de rocío es en grados centígrados 92
6.7. Experimento E. Imputación por variable Temp Out al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la la temperatura exterior es en grados centígrados 93

6.8. Experimento E. Imputación por variable Hi Temp al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la temperatura máxima es en grados centígrados 94
6.9. Experimento E. Imputación por variable Out Hum al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la humedad en porcentaje de humedad relativa 95

Índice de cuadros

5.1. Hiperparámetros y valores utilizados en la búsqueda aleatoria del modelo LSTM múltiple y multivariado. 76

6.1. Hiperparámetros y valores utilizados en la búsqueda aleatoria del modelo LSTM múltiple y multivariado. 77

6.2. Diseño experimental A. Métricas resultantes de la hiperparametrización de la arquitectura LSTM propuesta en la imputación de datos sobre las estaciónes 10, 11, 12 entrenadas con la estación 9. Comprendiendo 15 experimentos en total en esta sección 80

6.3. Diseño experimental A. Métricas resultantes en la imputación de datos sobre la estación 10 con el algoritmo EMB, MissForest y el algoritmo LSTM propuesto 80

6.4. Métricas de media y desviación estándar sobre los datos originales en la estación 10 con el algoritmo EMB, MissForest y el algoritmo LSTM propuesto 81

6.5. Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el algoritmo LSTM + Loung, Attention, muchos valores en negativo, no aprende, se descarta esta arquitectura 83

6.6. Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo Red convolutional + LSTM 83

6.7. Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo Transformer + LSTM 84
6.8. Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo LSTM Encoder-Decoder 84
6.9. Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo LSTM Bidireccional 84
6.10. Diseño Experimental C. Métricas resultantes en la imputación de datos sobre la estación 9 con el algoritmo EMB, MissForest y LSTM propuesto. Las métricas resultantes haciendo entrenamiento con la estación 8 e imputando sobre la estación 9. 86
6.11. Diseño Experimental A. Copia de la gráfica 6.2 para utilizar como referencia para comparar con la tabla anterior el diseño experimental A y el diseño experimental C. 86
6.12. Imputación de datos sobre las estaciones 10, 11, 12 entrenadas con la estación 9. Experimento D 87
6.13. Diseño experimental D. Métricas resultantes en la imputación de datos sobre la estación 10 con la arquitectura LSTM (EMB, MissForest utilizan las métricas obtenidas en el experimento A y se utilizan como referencia para comparación). 88
6.14. Diseño Experimental A. Copia de la gráfica 6.2 para utilizar como referencia para comparar con la tabla anterior el diseño experimental A y el diseño experimental D. 88
6.15. Diseño Experimental E. Métricas obtenidas al evaluar EMB, MissForest y LSTM propuesto en la imputación de la estación 10. 90
6.16. Diseño Experimental E. Métricas obtenidas al evaluar EMB, MissForest y LSTM con rangos de aceptación. 98

Introducción

A través del tiempo, el ser humano ha recopilado información para identificar patrones o realizar predicciones; esto con el fin de poder utilizar esta información de forma que le permita anticipar el futuro y tomar decisiones [11].

Las aplicaciones del análisis de las series de tiempo son muy diversas, incluyendo ingeniería de tráfico, economía, estudios de ambiente, ecología, investigaciones de energía y señales de procesamiento, entre otros [8]. Muchas de las decisiones importantes dependen de las predicciones: gerentes que evalúan la productividad futura para contratar, prestamistas que pronostican futuros pagos, médicos que forman diagnósticos y pronósticos, e incluso los comités de admisiones de doctorado en economía evalúan el éxito futuro de sus postulantes a la hora de otorgar las admisiones [1] [2], entre otros.

Sin embargo, los datos faltantes en las series de tiempo han sido un problema común y muy conocido, pero que a su vez es difícil de resolver [12]. No sorprende que existan faltantes de datos, pues no existen sistemas de observación ideales [13]. Estos casos de datos faltantes son producidos por diferentes razones: datos ruidosos, señales caóticas [10], fallas de comunicación en redes, pacientes de salud fuera de instituciones que son monitoreados y pierden comunicación con redes para envío de datos [13], problemas de mantenimiento en sensores [14], daños en equipos [15], etcétera.

En el área del clima, los faltantes de datos en las series de tiempo es una problemática muy común. Dentro de las opciones que los investigadores han utilizado para solventar

esta problemática se encuentran: la interpolación de datos y la imputación de datos (a ejemplificar más adelante), que buscan rellenar o adivinar esos faltantes de datos con la información disponible [7].

El problema con esos faltantes de información radica en que pueden llevar a resultados defectuosos o indeseables, incluyendo predicciones no precisas o problemas en la toma de decisiones [8].

Existen métodos para solventar las problemáticas de los datos faltantes en series de tiempo, como por ejemplo, construir alguna función que pase cerca de los datos conocidos o los más cercanos [13] para darle un valor aproximado al dato faltante. Sin embargo, normalmente el rellenado de los datos faltantes en las series de tiempo implica algunas suposiciones, pero estos deben ser rellenados en la forma más precisa posible, para evitar distorsiones de datos [13].

Desde la mitad de los ochentas del siglo anterior, métodos sofisticados han sido introducidos, incluyendo maximización de la expectativa (EM), métodos de estimación por peso, y en particular, el K-Nearest Neighbors (vecinos más cercanos), imputación múltiple e imputación bayesiana [16].

La meta de esta investigación es generar un estudio utilizando algoritmos de aprendizaje automático ("machine learning"), para imputación múltiple y multivariada en series de tiempo, que permita descubrir y explorar los resultados de diferentes algoritmos utilizados en el campo del aprendizaje profundo ("deep learning"). Además, pretende lograr la comprobación de los métodos de imputación múltiple y multivariada en los datos de la climatología costarricense, y verificar si benefician el llenado de los datos faltantes.

Marco Teórico

Las **series de tiempo** son una técnica estadística que resulta apropiada cuando los datos son medidos en forma repetida y en intervalos de tiempo iguales [17]. Los datos están compuestos por observaciones consecutivas tomadas en puntos temporales sucesivos. Existen dos tipos principales de series de tiempo basados en la longitud de los intervalos entre los puntos de tiempo de observación: series de tiempo espaciadas regularmente (de manera uniforme o igual) y series de tiempo espaciadas irregularmente (de manera desigual). El análisis de series espaciadas regularmente es menos compleja al emplear métodos de análisis convencionales, por ejemplo, el análisis de las tendencias temporales y descomposición, análisis de estacionalidad, entre otros. Sin embargo, es un desafío cuando las series de tiempo son irregulares, ya que la irregularidad de las series causa pérdida de información y hace más difícil capturar tendencias, correlaciones seriales y estacionalidad [18]. Muchas de estas fallas irregulares a la hora de capturar los datos en las series de tiempo, se deben a fallas en los dispositivos que recopilan la información [15].

Los datos de las series de tiempo pueden ser procesados por patrones de datos emergentes con el fin de predecir el comportamiento futuro para informar sobre políticas o uso de control de medidas. La extracción de información de series de tiempo depende de la confiabilidad de los datos y la integridad de la información. Al medir, recopilar o generar datos, los datos faltantes pueden ocurrir comúnmente como resultado de numerosos procesos, como errores de comunicación, fallas de los sensores que generan

datos o cortes de energía. La falta de datos en los estudios analíticos puede conducir a resultados defectuosos e indeseables, incluidas predicciones inexactas o malas decisiones en la formulación de políticas [8].

Existen diferentes tipos de datos: los **datos univariables** y los datos **datos multivariables**. En los datos univariables el prefijo "uni" significa "uno", en otras palabras, datos de una sola variable [19]. Se utilizan para estudiar el comportamiento de las variables de forma individual [20]. Algunos ejemplos: las edades de un grupo de personas, el tiempo de duración de un viaje en minutos, el peso de un grupo de cachorros, entre otros. Los datos multivariables se presentan cuando el análisis de los datos está basado en más de dos variables por observación. Usualmente los datos multivariables se usan con propósitos explicativos [21]. Es decir, es cuando el análisis de los datos está basado en dos o más variables por observación o, dicho de otra forma, cuando se observan dos o más variables en cada momento (en cada tiempo). Los datos multivariables pueden ser ejemplificados con un sensor climatológico que registre variables de clima, por ejemplo, rayos UV, temperatura, precipitación, etcétera.

Las **covariables** son variables independientes que junto a una o más variables sirven para explicar una variable de respuesta [22]. Por ejemplo, si pretendemos explicar las diferencias existentes en el nivel de cortisol en sangre por un grupo de tratamiento A / B, teniendo en cuenta el peso. La variable peso es una covariable [23].

Los **datos faltantes en series de tiempo** ocurren cuando solo pueden ser observados en ciertos periodos de tiempo [18]. Los datos de series de tiempo desigualmente espaciados (también llamados espaciados de manera desigual o irregular) ocurren naturalmente en muchos dominios industriales y científicos. Los desastres naturales como terremotos, inundaciones o erupciones volcánicas ocurren típicamente en intervalos de tiempo irregulares. En la astronomía observacional, las mediciones de propiedades tales como los espectros de los objetos celestes se toman según determinadas condiciones estacionales y climáticas, y la disponibilidad de los espacios de tiempo de observación. En los ensayos clínicos (o más generalmente, estudios longitudinales), el estado de salud de un paciente puede observarse sólo en intervalos de tiempo irregulares, y

generalmente se observan diferentes pacientes en diferentes momentos. Hay muchos más ejemplos en climatología, ecología, economía, finanzas, geología, meteorología y análisis de tráfico de red. [24].

Los principales factores responsables de los vacíos y las inconsistencias en las series de tiempo disponibles, que incluyen la ausencia temporal de observaciones, son: el uso de equipos diferentes, la poca e infrecuente calibración de los sensores, el mal funcionamiento del equipo que recopila la información, entre otros, que pueden variar de una colección de datos a otra [25, 10].

Para solventar el problema de los datos faltantes en series de tiempo se recurre a la metodología de **imputación de datos**. La imputación es el proceso de reconstrucción de los valores faltantes basados en los datos ya conocidos. Utilizando estadísticas para la imputación se pueden obtener buenos estimados de los datos no recopilados cuando las series de datos tienen una pequeña cantidad de vacíos. Sin embargo, es difícil predecir datos consecutivos en las series de tiempo [10].

Desde un punto de vista de estadístico, se define como el proceso de sustitución de un único valor razonable para una observación faltante. La imputación de datos faltantes en el área de la estadística, ha atraído la atención de investigadores en las décadas recientes. Muchas estrategias del proceso de rellenado de datos están inspiradas en estadísticas y aprendizaje automático (definido en la sección 2) que han sido desarrollados para solucionar el problema de datos faltantes [26].

En la investigación realizada en este documento, se pretende brindar una solución utilizando aprendizaje automático que sea una alternativa al proceso de rellenado de datos faltantes en series de tiempo en las predicciones climatológicas.

La imputación de datos puede ser **simple** o **múltiple**. La imputación simple de datos significa que sólo se construye un valor sustituto razonable para cada valor faltante, y lo interpola a la ubicación de los datos originales que resultan escasos para construir un conjunto de datos completo [7]. Por ejemplo, un valor razonable para estimar un valor faltante en una hora determinada de la temperatura diaria, podría ser el

promedio entre la hora anterior y la hora posterior a dicha hora faltante.

La imputación múltiple de datos, completa los datos faltantes en los datos utilizando un modelo predictivo que incorpora toda la información disponible de los datos observados junto con cualquier conocimiento previo. Se crean conjuntos de datos "completados" separados, donde los datos observados permanecen iguales, pero los valores faltantes se completan con diferentes imputaciones [27].

También existe la **imputación univariable** y la **imputación multivariable** de datos. Por definición de series de tiempo, univariable se entiende que son una secuencia de observaciones únicas o_1,o_2,o_3,...,o_n en puntos sucesivos en el tiempo t_1,t_2,t_3,...,t_n. Las observaciones univariables en series de tiempo pueden considerarse como una columna de observaciones, donde el tiempo es un factor implícito de la variable [15]. Las series de tiempo univariable son un desafío especial. La mayoría de los algoritmos estándar que mejor se desempeñan en la imputación de datos, se basan en las correlaciones entre los atributos para estimar los valores de los datos faltantes. En el caso de los datos univariables, no se pueden emplear atributos adicionales directamente. Los algoritmos univariados efectivos necesitan utilizar las características intrínsecas de series de tiempo [15].

En contraste con la imputación univariable, la imputación multivariable usa varios conjuntos de datos (denotados como m) para imputar los valores faltantes. Un valor faltante en su conjunto de datos original, es reemplazado por m valores plausibles imputados (como lo hace el algoritmo EMB utilizado en el presente estudio). Estos valores toman la incertidumbre de la imputación en consideración. Las estadísticas de interés se estiman a partir de cada conjunto de datos y luego se combinan en uno final [28]. Las series de tiempo multivariable son abundantes en muchas áreas de aplicación, como el mercado financiero, salud, meteorología, ingeniería de tráfico y climatología, entre otras. Las series de tiempo se utilizan ampliamente como señales para la clasificación/regresión en varias aplicaciones de las áreas mencionadas [14].

En el presente estudio se utiliza la imputación multivariable en climatología, a partir

de variables del clima que son capturadas por el ICAFE, con fin de evaluarlas en el proceso de siembra y cosecha del café en Costa Rica.

Dentro de las métricas más utilizadas en los modelos de regresión se encuentran: **error promedio absoluto (Mean Absolute Error MAE)**, **raíz de la desviación cuadrática promedio (Root Mean Square Error RMSE)** y el **coeficiente de determinación** (R^2).

El MAE mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección. Es el promedio sobre la muestra de prueba de las diferencias absolutas entre la predicción y la observación real, donde todas las diferencias individuales tienen el mismo peso [29].

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \mid y_i - \hat{y}_i|$$

La fórmula de MAE puede comprenderse como: n es el número de muestras observadas. Donde $\hat{y}_i$ y y_i (i=1,2,...,n) denotan los valores estimados y los valores observados, respectivamente.

Es una medida de uso frecuente de las diferencias entre valores (valores de muestra o población) predichas por un modelo y los valores observados.

El RMSE representa la raíz cuadrada del segundo momento muestral de las diferencias entre los valores predichos y los valores observados, es decir el promedio cuadrático de estas diferencias. Estas desviaciones se denominan residuales cuando los cálculos se realizan sobre la muestra de datos que se utilizó para la estimación y se denominan errores residuales (o errores de predicción) cuando los cálculos se realizan fuera de la muestra de datos que se utilizó para la estimación. El RMSE sirve para agregar las magnitudes de los errores en las predicciones varias veces en una sola medida de poder predictivo. RMSE es una medida de precisión que compara errores de pronóstico de diferentes modelos para un conjunto de datos en particular y no entre conjuntos de datos, ya que depende de la escala [30].

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(yi - \hat{yi}\right)^2}$$

Las fórmula de RMSE puede comprenderse como: n es el número de muestras observadas. Donde $\hat{y}_i$ y y_i (i=1,2,...,n) denotan los valores estimados y los valores observados, respectivamente.

El coeficiente de determinación o R2, se define como la proporción de la varianza total de la variable explicada por la regresión. El coeficiente de determinación, también llamado R cuadrado, refleja la bondad del ajuste de un modelo a la variable que pretende explicar [31].

$$r^2 = \left(\frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \tilde{y})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \tilde{y})^2}} \right)^2.$$

Es importante saber que el resultado del coeficiente de determinación, usalmente oscila entre 0 y 1, pero en algunos casos puede tomar valores negativos. Cuanto más cerca de 1 se sitúe su valor, mayor será el ajuste del modelo a la variable que estamos intentando explicar. De forma inversa, cuanto más cerca de cero, menos ajustado estará el modelo y, por tanto, menos fiable será [31].

Los métodos basados en **aprendizaje automático** son procedimientos sofisticados que generalmente consisten en crear un modelo predictivo para estimar los valores que van a sustituir los datos faltantes. Estos modelos de enfoque de estimación de datos faltantes están basados en información disponible en los conjuntos de datos. Si los valores observados contienen información útil para predecir los valores faltantes, el proceso de imputación puede utilizar esta información para dar con estimaciones de alta precisión [26].

Los métodos de aprendizaje automático han sido ampliamente estudiados y muchos trabajos han demostrado su utilidad en la predicción. Una de las ventajas de utilizar aprendizaje automático es que usualmente son más flexibles que los modelos estadísticos estándar, ya que pueden capturar en alto orden las interacciones entre los datos, lo cual resulta en mejores predicciones. En los modelos que utilizan redes neuronales

artificiales, por lo cual la interpretabilidad de los resultados es más compleja, aunque existen herramientas que facilitan la extracción y comprensión del conocimiento adquirido por estos modelos. [26].

Existen muchos métodos de aprendizaje automático para realizar predicciones e imputaciones de datos, como Support Vector Machines (SVM), redes neuronales artificales (ANNs), K-nearest neighbors (KNN), aprendizaje profundo (deep learning), entre otros [14].

El origen de los estudios en **redes neuronales artificiales** estuvo en la búsqueda de analogías cerebrales a partir de estudios fisiológicos iniciados a finales del siglo pasado [32]. Se trata de reproducir el esquema neuronal, tanto en sus unidades de proceso (neuronas), como en sus conexiones y modo de establecerlas [32].

Estos sistemas, inspirados en las neuronas densamente interconectadas del cerebro, tratan de imitar el aprendizaje humano, modificando la fuerza de las conexiones neuronales simuladas a partir de la experiencia [33].

El **aprendizaje profundo (deep learning)** es una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos. El aprendizaje profundo atrae mucha atención en tiempos recientes, y hay razones de peso para ello: está consiguiendo resultados que antes no eran posibles [34].

Con el aprendizaje profundo, un modelo informático aprende a realizar tareas de clasificación directamente a partir de imágenes, texto o sonido. Los modelos de aprendizaje profundo pueden obtener una precisión de vanguardia que, en ocasiones, supera el rendimiento humano. Los modelos se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales que contienen muchas capas [34]. Los algoritmos y arquitecturas de aprendizaje profundo son adecuados para abordar problemas con grandes volúmenes de datos y la variedad de análisis de Big Data [35].

La aplicación de técnicas de aprendizaje profundo ha llevado a un éxito dramático

y al desarrollo de soluciones del estado del arte en numerosos problemas del mundo real, como por ejemplo, visión por computadora, traducción automática, robótica, entre otros. En el contexto de predicción secuencial existe una gran gama de técnicas que han sido desarrolladas para solucionar el problema del procesamiento del lenguaje natural (NLP por sus siglas en inglés) en: secuencias genéticas, predicciones en el mercado de reservas, generación de música y un gran rango de otros dominios [36].

Mientras que muchos estudios discuten el uso de aprendizaje profundo para resolver directamente problemas de detección de anomalías e intrusiones, sólo algunos se enfocan en el problema de restaurar datos con pérdida o ruidosos [36].

Los métodos de aprendizaje profundo son muy prometedores para la previsión de series temporales, como el aprendizaje automático de la dependencia temporal y el manejo automático de estructuras temporales como tendencias y estacionalidad [37].

Las **redes neuronales recurrentes (RNN)** son un tipo de red neuronal potente y robusto, y pertenecen a los algoritmos más prometedores en uso porque es el único con memoria interna.

Debido a su memoria interna, los RNN pueden recordar cosas importantes sobre la entrada que recibieron, lo que les permite ser muy precisos al predecir lo que vendrá después. Es por eso que son el algoritmo preferido para datos secuenciales como series de tiempo, voz, texto, datos financieros, audio, video, clima y mucho más. Las redes neuronales recurrentes pueden formar una comprensión mucho más profunda de una secuencia y su contexto en comparación con otros algoritmos [38].

La mayoría de las RNN sufren de problemas con la memoria a corto plazo. Si una secuencia de datos es muy larga, tendrá problemas con sobrellevar la información de pasos en el tiempo anteriores a los posteriores. Durante el proceso de "back propagation", las RNN sufren del "vanishing gradient problem", que sucede cuando el valor gradiente se vuelve muy pequeño y no contribuye al aprendizaje de la red. Para solu-

cionar los problemas anteriores surgen los **Long Short-Term Memory (LSTMs)** como solución a los problemas de memoria a corto plazo, para ello los LSTM tienen mecanismos internos llamados puertas que pueden regular el flujo de información a guardar o olvidar, lo cual los hace ser una opción de red neuronal buena para procesar secuencias de datos grandes [3].

La figura 2.1 describe la forma interna de una neuronal LSTM. Su funcionamiento es similar al de las RNN, procesa los datos pasando información y propagándola hacia adelante, la diferencia principal radica en las operaciones ejecutadas en las celdas LSTM. Estas operaciones permiten a LSTM conservar u olvidar información [3].

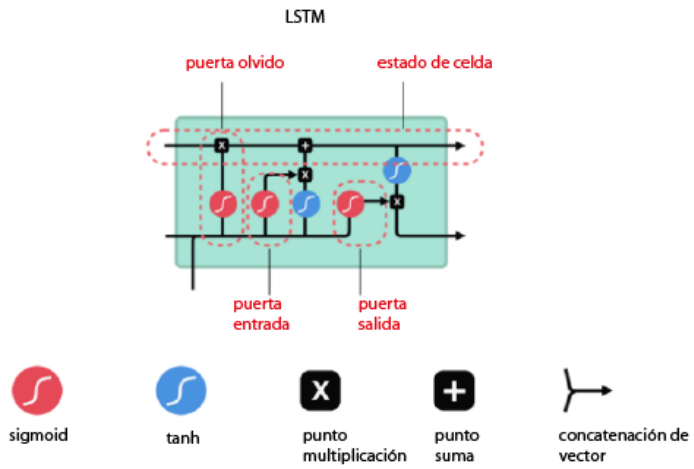


Figura 2.1: Representación de una neuronal LSTM: tomado de [3]

Los **LSTM Bidireccionales** es la técnica de permitir que cualquier red neuronal almacene información de secuencia en ambos sentidos, ya sea hacia atrás o hacia adelante. Nuestra entrada se ejecuta de dos maneras en bidireccional, distinguiendo un BiLSTM de un LSTM estándar. Podemos tener el flujo de entrada en ambas direcciones; para almacenar información pasada y futura en cualquier paso de tiempo. Sin embargo, los LSTM normales permiten el flujo de entrada en una dirección (hacia adelante o hacia atrás)[39].

Los LSTM bidireccionales son una extensión de los LSTM tradicionales que pueden

mejorar el rendimiento del modelo en problemas de clasificación de secuencias.

En problemas donde todos los pasos de tiempo de la secuencia de entrada están disponibles, los LSTM bidireccionales entrenan dos en lugar de un LSTM en la secuencia de entrada. El primero en la secuencia de entrada tal cual y el segundo en una copia invertida de la secuencia de entrada [39].

El **LSTM Encoder-Decoder** toma la secuencia de serie temporal como entrada (un paso de tiempo por celda LSTM) y crea una codificación de la secuencia de entrada. Esta codificación es un vector que consta de los estados ocultos y de celda de todas las celdas LSTM del codificador (encoder). Luego, el encoder se pasa al decodificador (decoder) LSTM como estados iniciales junto con otras entradas del decodificador para producir las predicciones (salidas del decodificador). Durante el entrenamiento del modelo, se configura la secuencia de salida de destino como las salidas del decodificador para el entrenamiento del modelo [4].

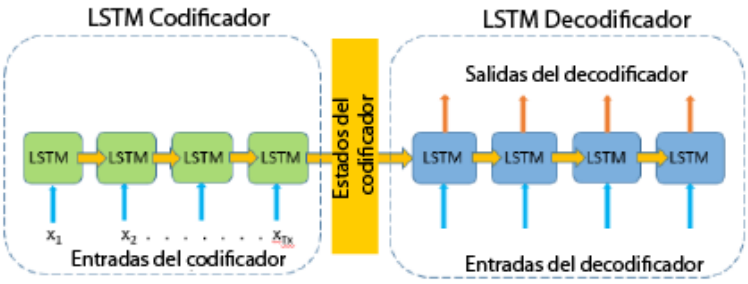


Figura 2.2: Encoder-Decoder [4]

El **Transformer** es un RNN que es capaz de considerar el contexto información, es más fácil de entrenar y más eficiente que LSTM, no requiere datos secuenciales para ser procesado en orden[40].

Redes Convolucionales (CNN) son un tipo de red neuronal que fue diseñada para manejar datos de imágenes de manera eficiente.

La capacidad de las CNN para aprender y extraer características automáticamente de los datos de entrada sin procesar se puede aplicar a los problemas de pronóstico de series temporales. Una secuencia de observaciones puede tratarse como una ima-

gen unidimensional que un modelo CNN puede leer y destilar en los elementos más destacados[37].

Las **redes con unidades recurrentes (GRU)** se comportan mejor con conjuntos de datos pequeños, pueden verse como una simplificación de las LSTM. Su simpleza permite que sean ampliamente usadas, dado que son mas fáciles de entrenar y son más rápidas en la inferencia que las LSTM[41].

Existen otras técnicas de aprendizaje supervisado como los **árboles de decisión**. Los arboles de decisión son comúnmente utilizados en problemas de clasificación y también en muchos negocios. Como su nombre lo indica, esta técnica de aprendizaje automático toma una serie de decisiones en forma de árbol. Los nodos intermedios (las ramas) representan soluciones. Los nodos finales (las hojas) nos dan la predicción que se busca. Los árboles de decisión pueden utilizarse para resolver problemas de clasificación o de regresión [42].

Un tipo específico de árboles de decisión son los **Random Forests o bosques aleatorios**. Están hechos de muchos árboles de decisión. Son conjuntos de árboles de decisión, cada árbol de decisión creado mediante el uso de un subconjunto de los atributos utilizados para clasificar una población determinada. Esos árboles de decisión votan sobre cómo clasificar una instancia dada de datos de entrada y el bosque aleatorio toma esos votos para elegir la mejor predicción. Esto se hace para evitar el sobreajuste, un defecto común de los árboles de decisión [43]. Un bosque aleatorio es un algoritmo de clasificación supervisado. Crea un bosque (muchos árboles de decisión), ordena sus nodos y se divide al azar [43]. Si ingresa un conjunto de datos de entrenamiento con objetivos y características en el árbol de decisión, formulará un conjunto de reglas que se pueden usar para realizar predicciones [43].

Para ejemplificar cómo funciona Random Forest en forma sencilla, suponga que se tiene un conjunto de datos que se compone de números de arriba a la derecha (ver 2.3). Se tiene dos 1s y cinco 0s (1s y 0s son las clases) y se desea separar las clases en características. Las características son el color (por ejemplo azul y rojo) donde las observaciones están en forma subrayadas o no subrayadas. El color resulta una

característica obvia pero solamente un 0 es color azul. Entonces, se puede usar la pregunta, ¿es rojo? y así continuar evaluando las demás características y de esta manera construir el árbol de decisión. La siguiente pregunta puede ser: ¿está subrayado? y de esta manera se ha desarrollado un árbol que es capaz de utilizar dos características para dividir los datos. [5]

Figura 2.3: Imagen ejemplo de Random Forest [5]

El modelaje a través de Random Forest, va mejorando las predicciones a mayor cantidad de árboles no correlacionados hayan en el modelo [5].

Otra técnica de aprendizaje supervisado, utilizada en esta investigación es el algoritmo de **maximización de la expectativa o expectation maximization (EM)**. El algoritmo EM, toma en cuenta el patrón de los datos faltantes para estimar los valores estadísticos que se calcularán para completar el conjunto de datos. La base de este método de imputación es interpolar estos valores faltantes con puntos de datos adyacentes, producir un par de valores de imputación y, finalmente, construir conjuntos de datos de muestra completos [7]. El algoritmo EM inicia asignando valores en forma aleatoria todos los parámetros para ser estimados. En forma iterativa alterna entre dos pasos: el paso de la expectativa 'E-step' y el paso de la maximización 'M-step'. En el 'E-step' se calcula la probabilidad esperada para los datos completos (llamada función Q) donde la expectativa es tomada del cálculo de la distribución condicional de las variables ocultas, dadas ciertas características de los parámetros y los datos observados (incompletos). En el paso 'M-Step', reestima todos los parámetros por medio de la maximización de la función Q. Una vez generados los valores de

los parámetros, se puede repetir el 'E- step' y otro 'M-step'. Intuitivamente, EM lo que hace es 'aumentar' iterativamente los datos 'adivinando' los valores de las variables ocultas y volver a estimar los parámetros asumiendo que los valores adivinados son los valores verdaderos [44].

Y una variación de EM es el algoritmo **maximización de la expectativa con boostrap o expectation maximization with boostrap (EMB)**, que fue desarrollado para resolver problemas de escasez de datos, al combinar las ventajas de enfoques de imputación múltiple y el algoritmo EM [7]. El algoritmo EMB basado en bootstrap hace que sea más rápido y robusto imputar muchas variables, incluyendo datos de series transversales, series de tiempo, entre otros [45]. La figura 2.4 describe el esquema del enfoque de imputación múltiple con el algoritmo EMB.

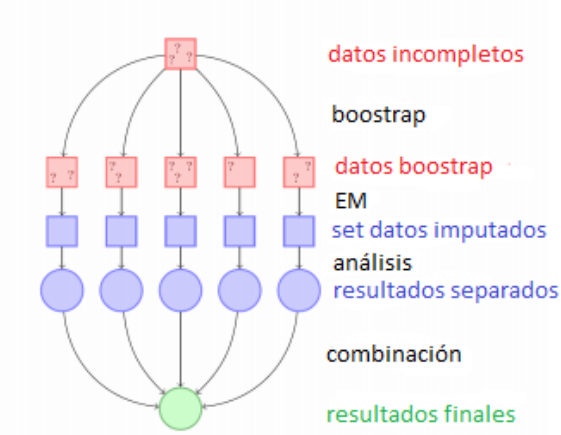


Figura 2.4: Esquema del algoritmo EMB. Fuente: tomado de [6]

EMB es un algoritmo general y efectivo que subyace a muchos algoritmos de aprendizaje automático. Requiere que el conjunto de datos de entrenamiento esté completo y que todas las variables que interactúan aleatoriamente estén presentes.

El algoritmo de maximización de expectativas es un enfoque para realizar la estimación de máxima verosimilitud en presencia de variables latentes. Lo hace estimando primero los valores de las variables latentes, luego optimizando el modelo y, finalmente, repite estos dos pasos hasta la convergencia. Es un enfoque efectivo y general y se usa más comúnmente para la estimación de densidad con datos faltantes, como

algoritmos de agrupamiento como el modelo de mezcla gaussiana[46].

La idea básica de utilizar Boostrap es producir una muestra aleatoria, de dónde se obtiene el muestreo con reemplazo del conjunto original de datos. Es utilizado para computar el estimado del parámetro en investigación (y esto produce la extracción de una muestra aleatoria y la computación del estimado) y se repite muchas veces para crear una distribución empírica que representa una buena aproximación al valor verdadero [47].

En esta investigación se pretende evaluar los métodos de aprendizaje supervisado como EMB y Random Forest y comprobar la efectividad de los algoritmos de aprendizaje automático a la hora de imputar datos sobre las series de tiempo y mostrar el impacto de algoritmos de aprendizaje automático en la predicción en series temporales que sean múltiples y multivariadas.

Descripción de la Investigación

3.1. Trabajos relacionados: "Datos faltantes en series de tiempo"(estado del arte)

La problemática de los faltantes en datos en las series de tiempo observadas ha sido un tema bien reconocido en diversos estudios [12, 15, 48] que muestran la realidad de los datos faltantes en observaciones a través del tiempo y buscan encontrar formas de solventar esta problemática (presente en muchos campos de estudio como biología, finanzas, ciencias sociales, industria de energía y observación del clima [15]) dados los diferentes factores que interfieren en la pérdida de estas observaciones, entre ellos se puede citar: negocios cerrados por un día, errores de comunicación en redes o sensores defectuosos [15], aleatoriedad e incontrolabilidad del viento [48], datos que solo pueden observarse en ciertos periodos de tiempo (condición de salud de pacientes o desastres naturales) [18], entre otros.

En la actualidad, existen muchos estudios relacionados con la necesidad de dar solución a los datos faltantes en las series de tiempo a través de la imputación de datos y utilizando técnicas de aprendizaje automático [12, 15, 49, 26, 48, 18, 16, 7], y, entre estos, muchos utilizan la técnica de imputación múltiple de datos [12, 48, 16, 7]. Según los autores Jerez et al [26], las técnicas de aprendizaje automático fueron los más adecuados en la imputación de valores faltantes y condujeron a una mejora significativa

de la precisión del pronóstico de cáncer de seno en comparación con los métodos de imputación basados en procedimientos estadísticos.

Estos estudios han utilizado muchos tipos de métodos diferentes para realizar este tipo de imputaciones de datos, por ejemplo, entre los más conocidos: K-Nearest Neighbors (KNN) [26, 48, 16, 12], Self-Organization Maps (SOM) y variaciones de este algoritmo utilizadas en [26, 12], Expectation Maximization (EM) [48] o Expectation Maximization con Boostrap (EMB) [7] (ampliado más adelante) y las variantes de EM como RegEM [12], métodos bayesianos [16], Random Forest [15, 8] (ampliado más adelante) y Redes Neuronales Artificiales (ANNs por sus siglas en inglés) [26].

Los resultados obtenidos por Miró et al en [12] demuestran que el algoritmo "Non-linear Principal Component Analysis"(NLPCA) y sus variantes (NLPCA + EOF-QM y NLPCA + EOF) son una alternativa poderosa en la imputación masiva de datos. En el artículo de Jerez et al [26], se concluye que la utilización de técnicas de aprendizaje automático son el mejor enfoque a la hora de imputar datos y obtener mejoras significativas en cuanto a la predicción de la exactitud. En el artículo de Liu et al [48], los resultados muestran que los métodos de imputación múltiple superan otros enfoques en el manejo de datos faltantes. En el artículo Quinteros et al [16], se determina que la imputación múltiple puede ser exitosa en la reconstrucción de un conjunto de datos con mejor rendimiento cuando se incluyen covariantes de otras estaciones.

En esta investigación se propone la utilización de datos múltiples (varias variables) y multivariables (varias estaciones geográficas con climatología similar) para la predicción de datos faltantes.

Para evaluar la efectividad de los métodos de imputación realizados en diferentes literaturas, los autores se han basado en métricas estadísticas para comprobar la efectividad de los algoritmos empleados. Dentro de las métricas más populares se encuentran Root Mean Squared Error (RMSE) [8, 48, 15, 12, 16, 49], Mean Absolute Error (MAE) [8, 48, 12, 16, 49, 50], Mean Absolute Percentage Error (MAPE) [15, 10, 12], Desviación Estándar [26], Mean Squared Error (MSE) [18, 50, 26], Coeficiente

de determinación (R2), entre otros [7, 16].

Es de interés del presente estudio ampliar sobre el algoritmo EMB utilizado en [7, 27], que muestra los siguientes resultados presentados por Chen et al [7], donde fue utilizado con el fin de resolver los problemas de escasez de lluvia registrada en las series de tiempo disponibles. En las figuras 3.1 y 3.2 se presentan los resultados obtenidos por EMB (Expectation Maximization Boostrap) y DA (Data Augmentation). Se puede observar que con el aumento en la tasa de escasez de 0 % al 60 %, en general, la variación en el algoritmo DA y el algoritmo EMB disminuyó de 106.50 a 58.31 y de 106.50 a 90.67, respectivamente. Sin embargo, la varianza en el algoritmo EMB no fue inferior a 86.00 y se ajustó mejor al valor observado que el algoritmo DA.

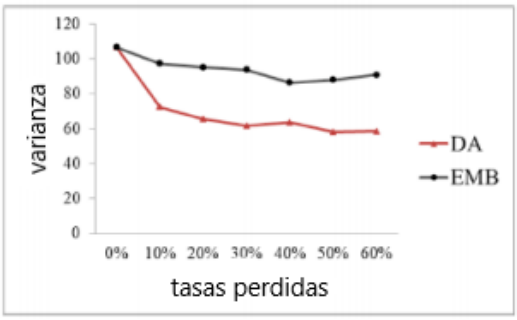


Figura 3.1: Comparación de los efectos entre los algoritmos de DA y EMB. Fuente: tomado de [7]

Table 1
Efectos de imputación de lluvia con 6 tasas faltantes

Tasas faltantes	DA	EMB
Ningún faltante	106.50	106.50
10%	72.19	97.28
20%	65.33	94.96
30%	61.51	93.65
40%	63.43	86.31
50%	58.10	87.92
60%	58.31	90.67

Nota: Los valores en la tabla son las variaciones de diferentes conjuntos de datos imputados y la varianza se calcula como: $\sigma^2 = \frac{\sum(x_i - X)^2}{n-1}$.

Figura 3.2: Efectos de la imputación de datos en seis tasas faltantes. Fuente: tomado de [7]

Por lo tanto, el conjunto de datos que fue interpolado por el algoritmo EMB fue mucho mejor que el interpolado por el algoritmo DA y estaba más cerca del valor observado [7]. También, el método de imputación EMB, se puede aplicar a la imputación de datos faltantes durante períodos de flujo alto, periódos de flujo normal o períodos de flujo bajo. Este hecho debe considerarse una ventaja importante del algoritmo EMB [7].

Este estudio sugiere que el algoritmo EMB puede resolver efectivamente el problema de escasez independientemente de las diferentes tasas faltantes y los diferentes períodos faltantes de datos de lluvia.

El otro algoritmo de interés es Random Forest utilizado en [8, 15], muestra que en las medias históricas, la interpolación lineal y los Random Forests obtuvieron resultados ligeramente mejores [8]. Como métrica se utiliza RMSE para evaluar la efectividad de la imputación de los datos. Cabe mencionar que en esta investigación el análisis principal no se centra en el comportamiento del algoritmo Random Forest. El primer ejemplo, ver figura 3.3, es la comparación de los algoritmos con el conjunto de datos de velocidad de tráfico en reconocimiento de patrones de letras. En el segundo ejemplo, ver figura 3.4, se comparan los algoritmos en los faltantes en patrones de flujo de agua. Y finalmente, el tercer ejemplo, ver figura 3.5, se utiliza en patrones faltantes en conjunto de datos Nottem. En las tres gráficas se puede evidenciar que Random Forest tiene un comportamiento aceptable (cerca de la media). Es por ello que se evaluará como algoritmo de comparación en el presente estudio.

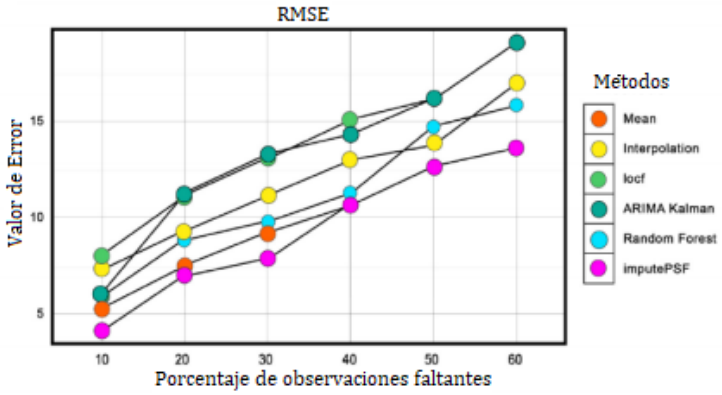


Figura 3.3: Porcentaje de observaciones faltantes en conjunto de datos de velocidad de tráfico en reconocimiento de patrones de letras.
Fuente: Tomado de [8]

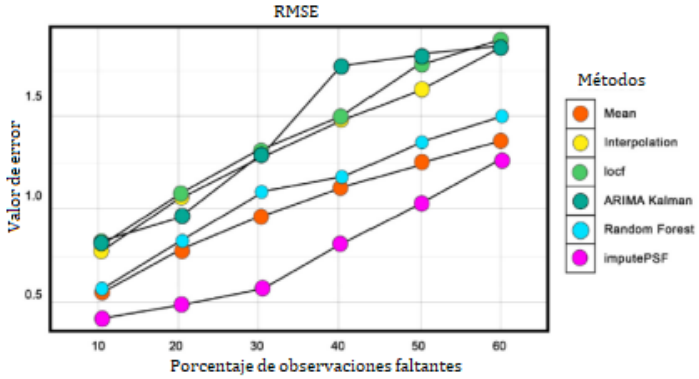


Figura 3.4: Porcentaje de observaciones faltantes en patrones de flujo de agua en series de tiempo. Fuente: Tomado de [8]

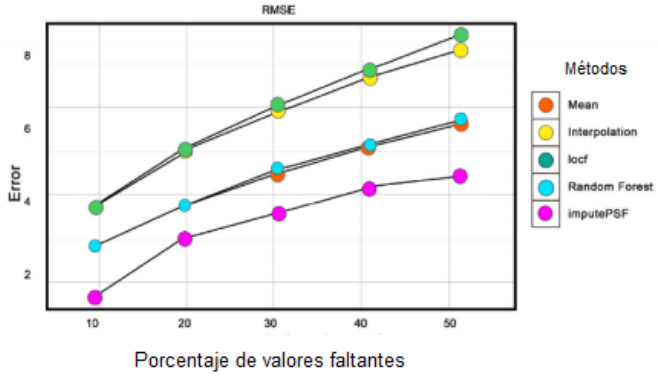


Figura 3.5: Porcentaje de observaciones faltantes en patrones del conjunto de datos Nottem. Fuente: Tomado de [8]

El algoritmo de aprendizaje profundo propuesto por Zhang et al en [9]; Sequence to Sequence Imputation Model (SSIM) es evaluado con otros métodos de imputación (EM, MF, ARIMA, SARIMA, MICE), para la evaluación se logran generar tres muestras de entrenamiento usando tres algoritmos diferentes de ventana deslizante. En la comparación de los algoritmos:

1. Generic Sliding Window (GSW) genera muestras con información histórica como entrada.

2. Sliding Window with Future information (SWF) también incluye información de pasos futuros en cada muestra.

3. El algoritmo Variable Length Sliding Window (VLSW) desarrollado por los autores para este estudio busca entrenar el modelo con conjuntos de datos más pequeños (lo que significa apertura para aplicar el modelo propuesto a otras arquitecturas).

El conjunto de datos utilizado en el estudio realizado por Zhang[9], utiliza múltiples variables como temperatura máxima, temperatura mínima, lluvia, evaporación, radiación, nitrato y otras más. Un detalle importante a notar es que el comportamiento del agua no sigue una distribución normal 3.6, lo cual puede impactar en el desempeño del modelo en forma significativa.

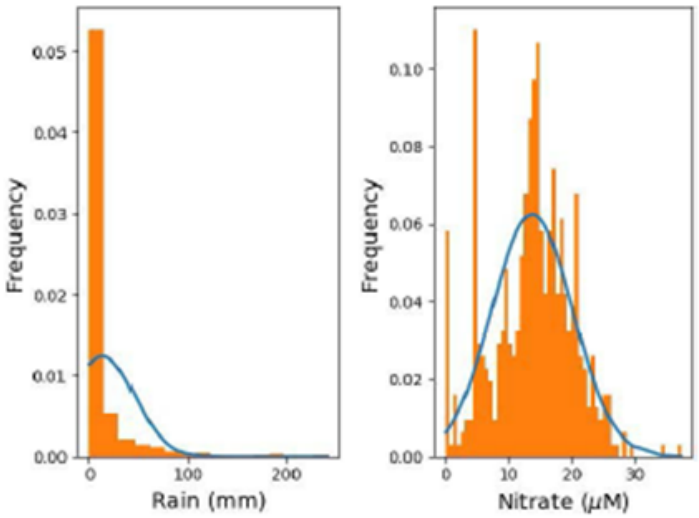


Figura 3.6: Distribución de la lluvia y concetraciones de nitrato durante Junio 2017 y Mayo 2018. Fuente: Tomado de [9]

La arquitectura desarrollada en SSIM se basa en modelos de secuencia - secuencia, que en años anteriores han demostrado efectividad en áreas de traducción, captura de imágenes y reconocimiento de voz. La mayoría de las implementaciones secuencia - secuencia utilizan redes neuronales recurrentes como Long Short Term Memory (LSTM).

Está claro que el SSIM ofrece el mejor rendimiento para cuatro métricas en los tres casos analizados. Por ejemplo, el SSIM con respecto a los otros métodos evaluados en el estudio (EM, MF, ARIMA, SARIMA y MICE) logra 2.87, 3.11 y 3.29 puntajes RMSE en los casos 1, 2 y 3, respectivamente, como se puede observar en la figura 3.7 (a) [9].

En comparación con los otros cinco métodos de imputación de datos comparativos, el SSIM puede mejorar el rendimiento de todas las métricas para los tres casos.

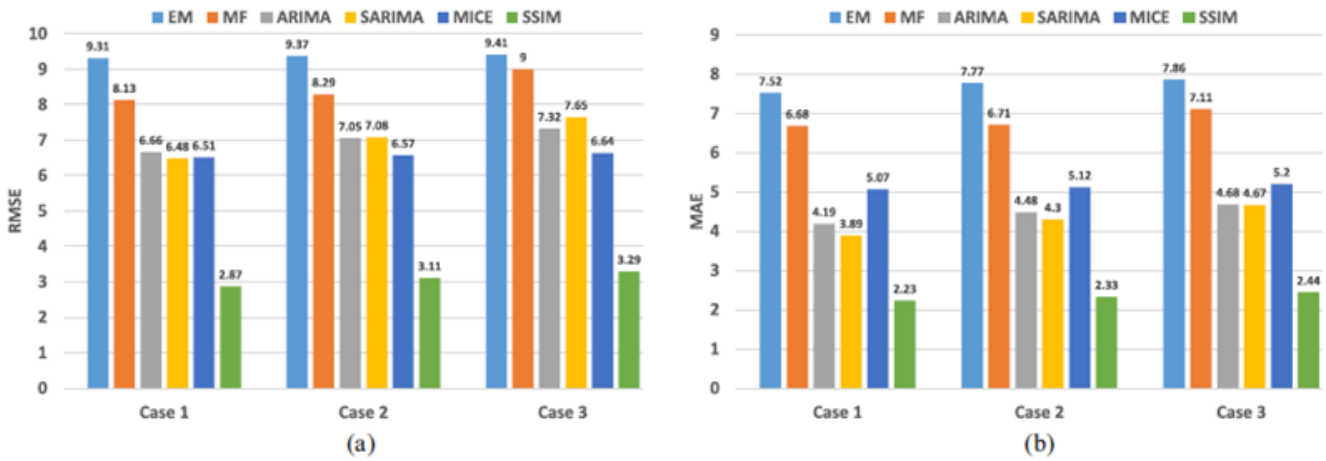


Figura 3.7: Evaluación de datos imputados para las métricas RMSE y MAE en 3 casos. Fuente: Tomado de [9]

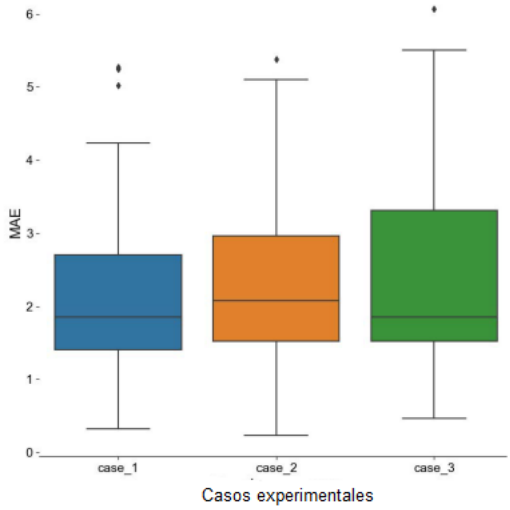


Figura 3.8: Gráfico Box-whisker para el MAE derivado de SSIM en los tres casos experimentales. Fuente: Tomado de [9]

La figura 3.8 representa la distribución del MAE del SSIM a través de los cuartiles. En todos los casos experimentales, aproximadamente el 25 % de los datos imputados señalados por el SSIM tienen un MAE inferior a 2 [9]. Además, más del 75 % de los puntos de datos imputados tienen un MAE inferior a 3 para el primer y el segundo caso. Cuando se imputan datos de series de tiempo en el tercer caso, el 75 % de los resultados generados por el SSIM todavía tienen un MAE menor que 3.2. Estos resultados demuestran claramente que el SSIM propuesto es capaz de imputar datos

de series temporales faltantes con una precisión satisfactoria para la mayoría de las muestras de prueba [9].

El algoritmo de aprendizaje profundo propuesto por Sucholutsky et al en [36], LTSM, logra predecir los eventos en un segmento (ver figura 3.9). El problema de omisión se presenta cada vez que se omite un evento en las predicciones, la secuencia completa de eventos predichos se desplaza hacia la izquierda, lo que provoca desajustes cada vez más grandes (comparar figura 3.9 y 3.10). El enfoque basado en LSTM (al ser capaz de detectar los problemas por omisión y darles solución) es, por lo tanto, lo suficientemente robusto para trabajar y restaurar no sólo datos con pérdida, sino también ruidosos [36]. Predecir múltiples pasos en el pasado es todo un reto, pero es importante para la restauración de datos faltantes, como múltiples eventos que han sido extraviados. Dos formas de predecir múltiples eventos en el pasado son considerados:

1. Incrementar el tamaño de la capa de salida por el factor de n para predecir directamente n eventos en el pasado, el problema de este enfoque es que cada vez que n cambia, una nueva red necesita ser inicializada y entrenada, lo cual requiere de muchos recursos.

2. Predecir únicamente una entrada cada vez, sin embargo el código fue modificado para usar las predicciones como entradas a la vez. De alguna manera, el modelo usa sus propios salidas, una cada vez, para hacer predicciones más largas.

	Evento Predicho	Evento cierto
1	2C6	2C6
2	5D7	5D7
3	B0	B0
4	224	224
5	B2	B2
6	20	20
7	B4	B4
8	25	25
9	22	22
10	23	23

Todos los eventos en este segmento son predichos correctamente y calzan con sus contrapartes.

Figura 3.9: Ejemplo de alineamiento propio. Fuente: Tomado de [10]

	Predicted Event	True Event
1	B4	B4
2	25	25
3	22	22
4	23	23
5	B0	**340**
6	320	B0
7	B2	320
8	2D0	B2
9	2C4	2D0
10		2C4

Figura 3.10: Ejemplo de valor omitido. Fuente: Tomado de [10]

El modelo LSTM fue suficientemente robusto para restaurar no sólo datos faltantes, sino también datos con ruido.

En la investigación realizada por Cao en [14], la mayoría de los métodos basados en RNN, excepto GRU-D, demuestran un rendimiento significativamente mejor en las tareas de imputación que los algoritmos non-RNN (Redes Neuronales no Recurrentes). Se hace hincapié en que GRU-D no imputa el valor faltante explícitamente. M-RNN utiliza un procedimiento de imputación explícito y logra resultados de imputación notables. El modelo BRITS supera significativamente a todos los modelos de referencia (para referencia ver figura 3.11). Los resultados de los experimentos indican que BRITS demuestra resultados más precisos tanto para la imputación como para la clasificación/regresión que los métodos de vanguardia. Es importante aclarar que la capa recurrente en la arquitectura de BRITS está basada en LSTM como RNN [14].

Es de interés, para el presente estudio, evaluar la imputación en series de tiempo utilizando los algoritmos de Expectation Maximization with Boostrap (EMB), MissForest (un tipo de Random Forest) y una arquitectura basada en Long Short Term Memory (LSTM) para evaluar las imputaciones con múltiples variables y múltiples estaciones que permitan evaluar la efectividad del aprendizaje automático utilizando LSTM en la imputación de datos faltantes en comparación a los algoritmos estadísticos tradicionales que han demostrado gran efectividad a la hora de imputar datos faltantes en series de tiempo.

Comparación de desempeño para tareas de imputación (en MAE) MRE%))

	Método	Calidad aire	Salud	Actividad Humana
Non-RNN	Mean	55.51 (77.97%)	0.720 (100.00%)	0.767 (96.43%)
	KNN	29.79 (41.85%)	0.732 (101.66%)	0.479 (58.54%)
	MF	27.94 (39.25%)	0.622 (87.68%)	0.879 (110.44%)
	MICE	27.42 (38.52%)	0.634 (89.17%)	0.477 (57.94%)
	ImputeTS	19.58 (27.51%)	0.390 (54.2%)	0.363 (45.65%)
	STMVL	12.12 (17.40%)	/	/
RNN	GRU-D	/	0.559 (77.58%)	0.558 (70.05%)
	M-RNN	14.24 (20.43%)	0.451 (62.65%)	0.248 (31.19%)
Ours	RITS-I	12.73 (18.32%)	0.395 (54.80%)	0.240 (30.10%)
	BRITS-I	11.58 (16.66%)	0.361 (50.01%)	0.220 (27.61%)
	RITS	12.19 (17.54%)	0.300 (41.89%)	0.248 (31.21%)
	BRITS	**11.56 (16.65%)**	**0.281 (39.14%)**	**0.219 (27.59%)**

Figura 3.11: Comparación de rendimiento de las tareas de imputación.
Fuente: Tomado de [14]

3.2. Objetivos

3.2.1. Objetivo general

Evaluar una estrategia de imputación de datos, múltiple y multivariada, de valores faltantes en series de tiempo en el campo meteorológico utilizando aprendizaje automático.

3.2.2. Objetivos específicos

Los objetivos específicos de este proyecto son los siguientes:

1. Evaluar varios métodos conocidos de imputación de datos múltiple y multivariada con el fin de medir su nivel de predicción con las métricas acordadas.
2. Proponer una estrategia de imputación de datos múltiple y multivariada.
3. Realizar un análisis comparativo del desempeño obtenido al realizar imputaciones múltiples y multivariadas en las series de tiempo con los métodos propuestos y la estrategia presentada.

3.3. Hipótesis

Con base en la justificación del problema y en los objetivos del proyecto, se define la siguiente hipótesis:

Utilizando la estrategia propuesta, basada en el aprendizaje profundo, para la imputación múltiple y multivariada de series de tiempo, es posible reducir el RMSE respecto a las otras técnicas utilizadas como comparación, EMB y Random Forest, en al menos un 5 % del valor obtenido en dicha métrica en las técnicas supramencionadas en la imputación de los conjuntos de datos propuestos.

3.4. Métricas

La determinación de las métricas a considerar para la evaluación del desempeño de la estrategia propuesta se basan en las métricas comúnmente utilizadas en los artículos presentados en el estudio del estado del arte [12, 48, 18, 16]. Las métricas a emplear serían:

1. Mean Absolute Error (MAE)
2. Root Mean Square Error (RMSE)
3. Coeficiente de determinación (R2)
4. Tiempo de ejecución: Tiempo que le toma al algoritmo utilizado en realizar la imputación de datos.

Previamente detalladas en la sección 2.

Diseño Experimental

4.1. Descripción de experimentos

Este capítulo expone los pasos desarrollados para implementar la propuesta *Imputación múltiple y mulivariada de valores faltantes en series de tiempo del campo meterológico utilizando aprendizaje automático.* Se describe el diseño de experimentos planteados basados en los lineamientos desarrollados en el libro de Douglas C. Montgomery [51].

4.1.1. Descripción del Problema

Desarrollar una propuesta de imputación múltiple y multivariada de series de tiempo que, utilizando aprendizaje automático, mejoren los resultados obtenidos con los algoritmos propuestos como punto de comparación.

4.1.2. Variables de Respuesta

Las variables de respuesta corresponden a las métricas previamente detalladas en la sección 2 de este documento:

1. MAE

2. RMSE

3. R2

4. Tiempo de ejecución

4.1.3. Factores

Se enumeran todos los posibles factores que puedan influenciar el experimento. Selección de factores:

Algoritmos: Los algoritmos seleccionados de la literatura analizada en esta investigación, en la sección 3.1, se aplicarán a los conjuntos de datos seleccionados de ICAFE.

- Algoritmo EM basado en bootstrap. Se utilizará un algoritmo para la imputación múltiple de los datos faltantes. Este algoritmo permite reducir la parcialidad e incrementar la eficiencia. Está habilitado por el boostrap basado en el algoritmo EM que hace más rápida y robusta la imputación de varias variables, incluyendo series de tiempo.

- Random Forest (MissForest): Se utilizará una implementación del algoritmo base Random Forest, denominada MissForest, la cual realiza una imputación no paramétrica que se puede aplicar a varios tipos de variables. Construye un modelo Random Forest para cada variable. Después, usa el modelo para predecir valores faltantes en la variable con ayuda de otros valores observados.

- Propuesta de la presente investigación.

Conjuntos de datos: Se utilizaron los siguientes datos reales facilitados por ICAFE: estaciones climatológicas pertenecientes a una misma zona geográfica identificadas por los números 9, 10, 11 y 12. Cada una de ellas cuenta con las variables de precipitación, temperatura máxima, temperatura exterior, punto

de rocío y humedad exterior. Estas variables fueron preseleccionadas con la intención de que compartieran algún grado de correlación entre ellas. La cantidad de datos disponibles para cada estación comprenden desde 01-12-2013 hasta el 30-04-2015 para un total de 12382 observaciones (registradas por hora) para cada estación.

Conjuntos de entrenamiento, validación y pruebas: Consiste en dividir los datos en un 80 % para entrenamiento, 10 % para validación y 10 % para pruebas.

Cantidad de datos faltantes en los diseños experimentales: Consiste en generar series de tiempo con faltantes de datos para ser evaluados por igual en los diferentes algoritmos al 1 %, 3 %, 5 %, 10 % y 20 %, a partir de los datos originales previamente trabajados.

4.1.4. Selección del diseño experimental

Para el diseño experimental se realiza entrenamiento con los datos de la estación 9 y se imputa y valida sobre las estaciones 10, 11 y 12, dando como resultado 15 experimentos para evaluar la arquitectura LSTM propuesta.

Estación Imputada	Algoritmo	% Train, Val & Test	% Datos faltantes
9	LSMT propuesto	80/10/10	1,3,5,10 y 20
11	LSMT propuesto	80/10/10	1,3,5,10 y 20
12	LSMT propuesto	80/10/10	1,3,5,10 y 20

También se comparan los algoritmos EMB, MissForest y la arquitectura LSTM propuesta entrenando con la estación 9 e imputando sobre las estaciones 10, 11 y 12 (se utilizan 5 experimentos generados en el paso anterior) y se comparan con los 10 experimentos nuevos de los resultados EMB y MissForest, dando como resultado 10

experimentos más. Para un total de 25 experimentos en esta sección.

Estación Imputada	Algoritmos	% Train, Val & Test	% Datos faltantes
10	EMB	80/10/10	1,3,5,10 y 20
11	MissForest	80/10/10	1,3,5,10 y 20
12	Algoritmo desarrollado	80/10/10	1,3,5,10 y 20

Las métricas descritas en la sección 3.4 serán utilizadas para analizar cada uno de estos experimentos como variables de respuesta en el análisis de los resultados.

4.1.5. Análisis estadístico

Para validar si las distribuciones obtenidas como resultado de los experimentos mostraron diferencias de medias significativas de acuerdo a la estrategia utilizada, se emplearon bibliotecas de sklearn metrics [52] en Python para realizar el análisis estadístico según las métricas mencionadas en la sección 4.1.2.

4.2. Detalles de los experimentos

A lo largo de esta sección, se profundiza en los detalles técnicos y del proceso relacionados a la preparación, implementación y ejecución de los experimentos realizados.

4.2.1. Ambiente de desarrollo

Esta sección detalla los requerimientos necesarios para lograr el preprocesamiento, implementación, ejecución y validación de los experimentos, que facilitaron el desarrollo y la configuración de los métodos a utilizar. A continuación, se describen los aspectos técnicos requeridos para lograr la ejecución de los experimentos.

Se propone el uso de las siguientes tecnologías:

1. Lenguaje de programación Python: para el uso de la biblioteca Keras, missForest y otras tareas de propósito general. Bibliotecas de Python: pandas, numpy, random, math, statsmodels, scipy, sklearn, matplotlib, IPython, wandb, tensorflow, kerastuner, keras, math, seaborn.
2. Lenguaje de programación R: para el uso de la biblioteca de Amelia. Bibliotecas de R: Amelia, Rcpp
3. Gestor de versiones de código: GIT
4. Notebooks de Jupyter en Cloud: Google Colab
5. Sistema Operativo y hardware mínimo: Las pruebas fueron realizadas en Google Colab, por lo tanto, se requiere una computadora con acceso a internet y una cuenta gratuita en Google Colab. Las especificaciones técnicas de la máquina utilizada para realizar el presente estudio son: Windows HP Pavilion con un Intel Core I3, 8GB RAM de 2.0 GHZ procesador.

4.2.2. Conjuntos de datos

El conjunto de datos utilizados en esta investigación fueron proporcionados por el Instituto del Café de Costa Rica (ICAFE), que cuenta con una serie de sensores en diferentes zonas alrededor del país en donde se monitorean múltiples variables de la climatología costarricense. Sin embargo, por diversas razones, los datos no siempre se han podido recolectar de forma efectiva.

El conjunto de datos de ICAFE está compuesto por 32 estaciones (a noviembre del 2019), en donde cada estación cuenta con múltiples variables, y para efectos de este estudio se delimitará técnicamente a las variables de temperatura exterior (en grados centígrados), temperatura máxima (en grados centígrados), humedad exterior (porcentaje de humedad relativa), punto de rocío (en grados centígrados) y precipitación (en milímetros cúbicos). Es importante aclarar que cada estación cuenta con 2 senso-

res, uno a la intemperie y otro ubicado en una bodega, por lo cual la humedad interior y exterior pueden no estar correlacionadas por factores como el aire acondicionado o ventiladores en bodega.

Para el entrenamiento y la prueba de los modelos desarrollados en esta investigación se utilizaron cuatro conjuntos de datos (detallados en el diseño experimental 4.2.2) provenientes de estaciones climatológicas ubicadas en la zona de Los Santos, con características muy similares entre sí (en su ubicación geográfica y características climatológicas). En la siguiente gráfica 4.1, se muestra el análisis de los datos realizado a las 32 estaciones facilitadas por ICAFE, las columnas representan el año, las filas el número de estación y los espacios en blanco representan días faltantes en la serie de tiempo. Cuando el espacio en blanco entre los datos es pequeño, puede valorarse si lo que faltan son horas o días que podrían rellenarse con el fin de lograr una serie de tiempo más completa (bajo criterios explicados más adelante). Cuando el espacio en blanco es mayor a 2 pixeles, según la gráfica, puede entenderse que faltan más de 2 días consecutivos en la serie de tiempo y por ende se debe valorar si utilizar esa franja de tiempo o incluso descartar esa estación para su análisis. Las estaciones están agrupadas de acuerdo con la zona geográfica a la que pertenecen. Las zonas geográficas se distinguen entre ellas por colores diferentes. Algunas de las celdas en color blanco pueden ser trabajadas para realizar una imputación de datos suponiendo que sean días u horas que manejan una misma climatología y que los extremos de los datos que los rodean muestren que no ha cambiado en forma significativa el clima en ese rango de datos ausente. La idea de rellenar esos datos es convertir los espacios en blanco al color de su zona, con el fin de producir un conjunto de datos más robusto para utilizar en los entrenamientos.

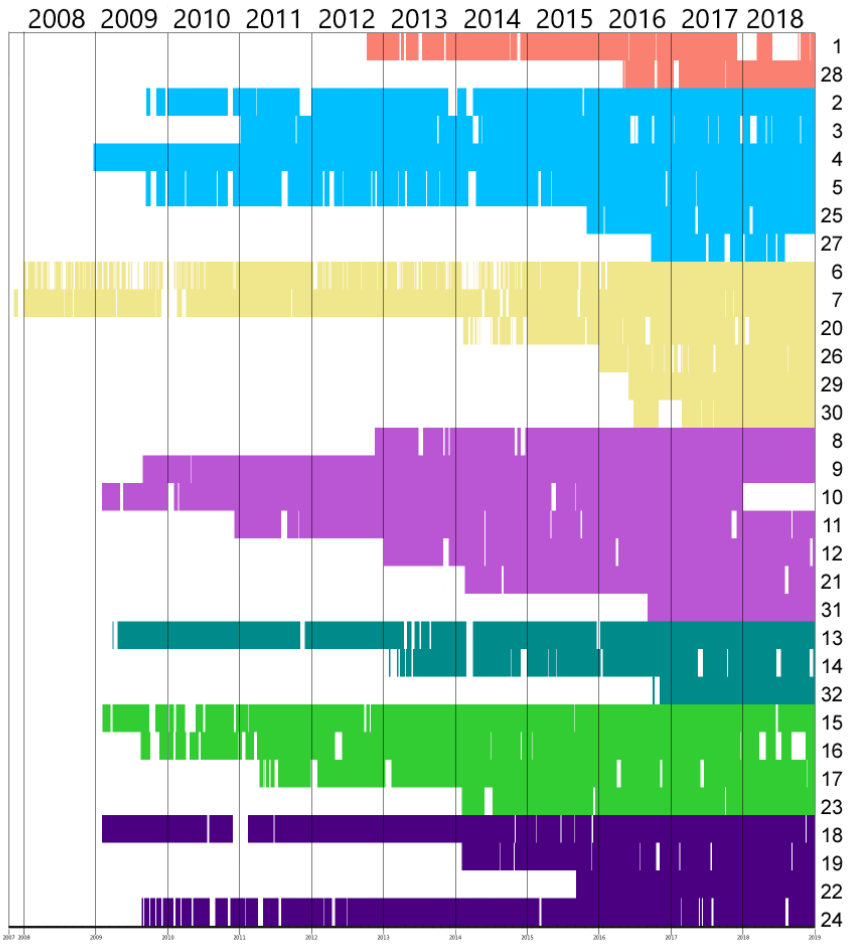


Figura 4.1: Presencia de datos en las 32 estaciones de ICAFE

4.2.3. Preprocesamiento de datos

Con el propósito de preprocesar el conjunto de datos, muchos factores debieron analizarse con el fin de obtener un conjunto de datos con características similares entre las estaciones y su periocidad. El proceso de preprocesamiento implica realizar una serie de transformaciones al contenido de cada estación con el fin de disminuir el ruido presente en los datos en las series de tiempo y así lograr que el conjunto de datos sea lo más uniforme posible. Los pasos de preprocesamiento utilizados en la propuesta final son:

1. Se toman las 32 estaciones a analizar y se generan nuevas estaciones con las 5 variables de interés, mencionadas anteriormente: temperatura exterior (Temp Out), temperatura máxima (Hi Temp), humedad exterior (Out Hum), punto de rocío (Dew Pt.) y precipitación (Rain).

2. Para cada una de las estaciones se proceden a generar conjuntos de datos con las mismas periocidades en las series de tiempo, es decir cada 15 minutos, 30 minutos, 45 minutos y cada 60 minutos con el fin de buscar la combinación más adecuada en cantidad de datos y periocidad según las estaciones disponibles en el estudio.

3. Una vez analizadas las 32 estaciones se procede a analizar la presencia de los datos en las 32 estaciones, dos requisitos importantes deben cumplirse al seleccionar un conjunto de datos: se requieren 2 o más estaciones de una misma zona geográfica y además deben compartir el mismo intervalo de tiempo con datos. Esto con el fin de escoger las estaciones con características similares (en su ubicación geográfica) y que contengan una cantidad poco significativa de valores faltantes (preferiblemente celdas con espacios mínimos en blanco), pues el grupo de datos a utilizar debe ser lo más completo posible para evitar introducir "outliers"(observaciones que se salen del comportamiento normal de los datos que a la vez pueden introducir ruido a los algoritmos de aprendizaje automático) en el proceso de rellenado manual y también evitar que cualquier proceso

de rellenado manual o automático (si es utilizado frecuentemente para rellenar faltantes) agregue el riesgo de perder el comportamiento real de los datos.

4. En conjunto con una persona experta de ICAFE, se procede a analizar el conjunto de los datos extraídos: la identificación de las estaciones con climatología similar (se determina que la zona de Los Santos, representada en color morado, cumple estas características) y que además contengan datos completos (o casi completos), la definición de criterios de llenado o imputación de datos según la cantidad de horas faltantes (sólo se rellenan espacios en blanco menores iguales a 4 horas) y la temporada en que se encuentran dichos datos faltantes. Todo lo anterior se realizó con el fin de definir las estaciones y los intervalos en las series de tiempo que van a ser la base para el entrenamiento de los algoritmos de aprendizaje automático.

5. Con las estaciones preseleccionadas se procede a analizar la cantidad de datos según las periocidades generadas. Para este estudio se determinó que la periociodidad de los datos de cada 60 minutos, en la zona de Los Santos (en color morado 4.1), en el intervalo de años de diciembre 2013 a abril 2015 (en el siguiente punto se proporciona el detalle de cómo se obtuvieron), produce una cantidad de 12382 datos por estación para poder entrenar los algoritmos.

6. Se procedió a realizar rellenado automático de datos en donde se realiza un barrido por todas las estaciones 9, 10, 11 y 12 de la zona de Los Santos. Este proceso consiste en rellenar cualquier faltante de datos menor o igual a 4 horas de datos en un día en horas consecutivas. Si el rango de horas faltantes es mayor se procede a generar arhivos de intervalos indicando el día inicial y la hora inicial y la fecha final con su hora final, de esta manera se identifica a partir de cuándo no se pudo rellenar los datos, pues el "hueco" faltante era mayor a las 4 horas consecutivas. Para ejemplificar el proceso la siguiente imagen muestra cómo se producen los intervalos en la estación 9.

Figura 4.2: Archivos de intervalos generados de la estación 9, se muestra el rango de tiempo de datos faltantes

7. En este paso, se procede a analizar los archivos de intervalos generados en cada estación, para verificar si es viable realizar algún tipo de rellenado manual de datos. Se analizan los días que colindan con el "hueco" y a través de observación del comportamiento de todas las variables, se decide si rellenar o no los datos con los datos de alguno de los extremos, tomando el comportamiento de días previos, y revisando que las variables compartan un comportamiento similar a los días que colindan directamente con los faltantes en las horas que se requieren rellenar. Todo esto según las recomendaciones dadas por el experto de ICAFE mencionado en el punto tras anterior. En caso de que se descarte realizar el rellenado de datos manual, ese intervalo es descartado como opción para el conjunto de datos que se busca generar.

8. Con los datos rellenados en las estaciones preseleccionadas, y con la periocidad deseada, se analiza el conjunto de los datos con el fin de determinar los subgrupos de análisis de los datos y sus cantidades, y de esta manera determinar cuál será el subconjunto de datos a seleccionar para entrenar los algoritmos.

Un ejemplo del preprocesamiento realizado se puede apreciar en la gráfica 4.3, donde

se muestran los datos originales para la zona de Los Santos en las estaciones 8, 9, 10, 11, 12, 21 y 31, en donde existen algunas líneas delgadas en color blanco, indicando que hay datos que pueden ser rellenados para generar una continuidad de los datos más amplia. Cuando las líneas son delgadas indican poca ausencia de datos en días. Siguiendo los pasos detallados anteriormente se determina que las estaciones 9, 10, 11 y 12, con fechas entre diciembre 2013 y abril del 2015 nos brindan la posibilidad de tener 4 estaciones con condiciones geográficas similares y cada una con 12372 observaciones por cada hora todos los días desde diciembre 2013 a abril del 2015, para poder utilizarlas con confianza en el diseño experiemental.

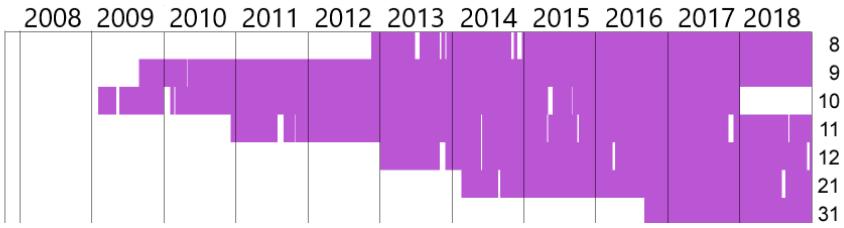


Figura 4.3: Filas: Identificación de la estación, columnas: año, y en cada celda la presencia de los datos originales en intervalos de cada día, facilitados por ICAFE

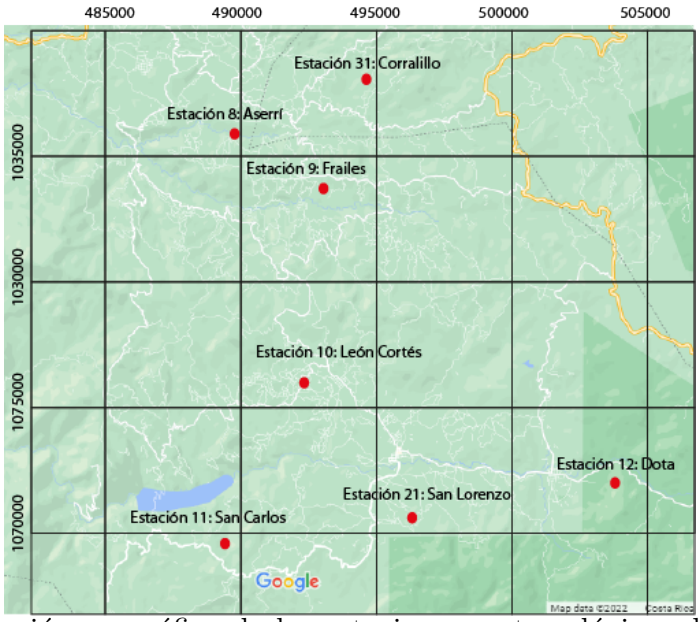


Figura 4.4: Ubicación geográfica de las estaciones metereológicas de la Zona de los Santos, Costa Rica.

En la gráfica 4.5, se pueden apreciar seis subconjuntos de datos, representados por los colores amarillo, rojo, azul, verde, negro y gris, estos subconjuntos de zonas se evaluaron durante el preprocesamiento de datos con el fin de encontrar un subconjunto de datos lo suficientemente grande que agrupe varias estaciones y que además aglomere la mayor cantidad de datos posibles y utilice lo menos posible el proceso manual de rellenado de datos.

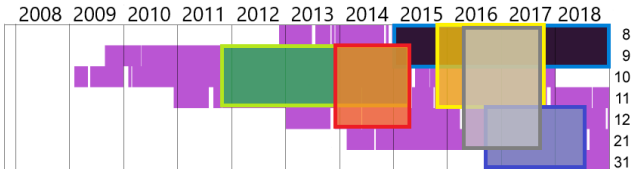


Figura 4.5: Muestra diferentes subconjuntos de datos representados por diferentes colores

El análisis de los datos realizado en la estación de Los Santos 4.5, en forma preliminar nos mostró que, aunque habían muchas zonas candidatas a ser evaluadas para el presente estudio, al realizar el análisis de forma minuciosa en cuanto a la cantidad de horas faltantes en mismo un día (que pueden ser no consecutivas pero a la vez tener muchos faltantes) y en cuanto a la cantidad de días consecutivos con pérdida de datos (que, aunque pueden rellenarse, introducen mucho ruido a los datos), se concluyó que seleccionar la zona roja con las estaciones 9, 10, 11 y 12 comprendidas desde diciembre 2013 a abril 2015 es la opción más viable.

A continuación se detalla la cuantificación de datos faltantes por estación (que cada estación contiene un total de 61860 valores):

- Estación 9: 85 valores faltantes fueron rellenados en forma automática y manual equivalentes a 17 horas.

- Estación 10: 310 valores faltantes fueron rellenados en forma automática, no hubo relllanado manual, equivalentes a 62 horas.

- Estación 11: 615 valores faltantes fueron rellenados en forma automática y manual, equivalentes a 123 horas.

- Estación 12: 615 valores faltantes fueron rellenados en forma automática y manual, equivalentes a 123 horas.

Se notó que, cuando las variables se agrupan y se comparan las gráficas de las estaciones, estas muestran tendencias similares en el periodo de tiempo analizado. Por ejemplo cuando son agrupadas por variable temperatura exterior 4.6, humedad exterior 4.7, temperatura máxima 4.8, las estaciones 9 y 11 y las estaciones 10 y 12 muestran similitudes en cuanto al comportamiento de estas variables, el punto de rocío 4.9 muestra una tendendcia similar en las 4 estaciones. Y en cuanto al gráfico de precipitación4.10 las estaciones 9, 10 y 11 comparten una tendencia similar, siendo la estación 12 la que difiere respecto a las demás estaciones.

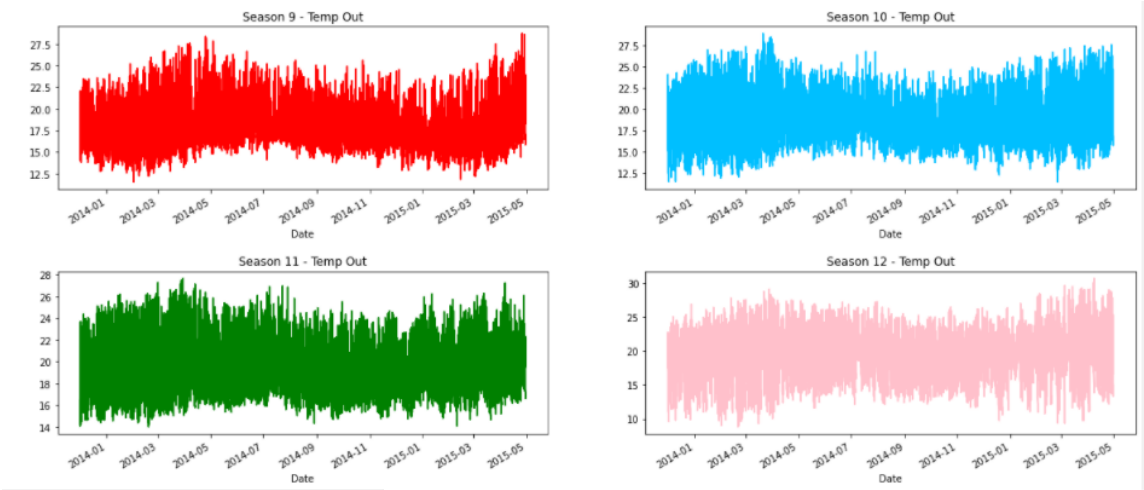


Figura 4.6: Temperatura Exterior en las estaciones 9, 10 11 y 12.

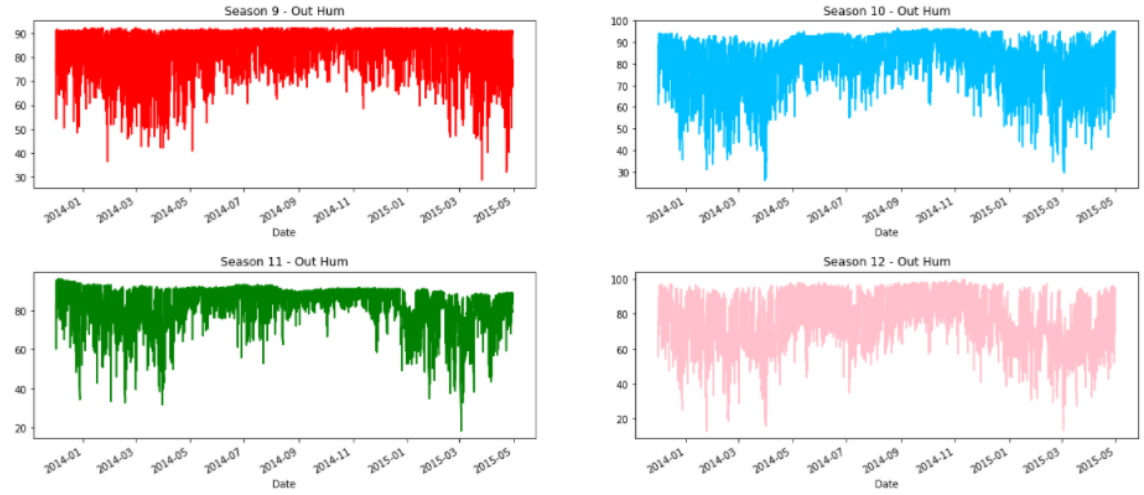


Figura 4.7: Humedad exterior en las estaciones 9, 10 11 y 12.

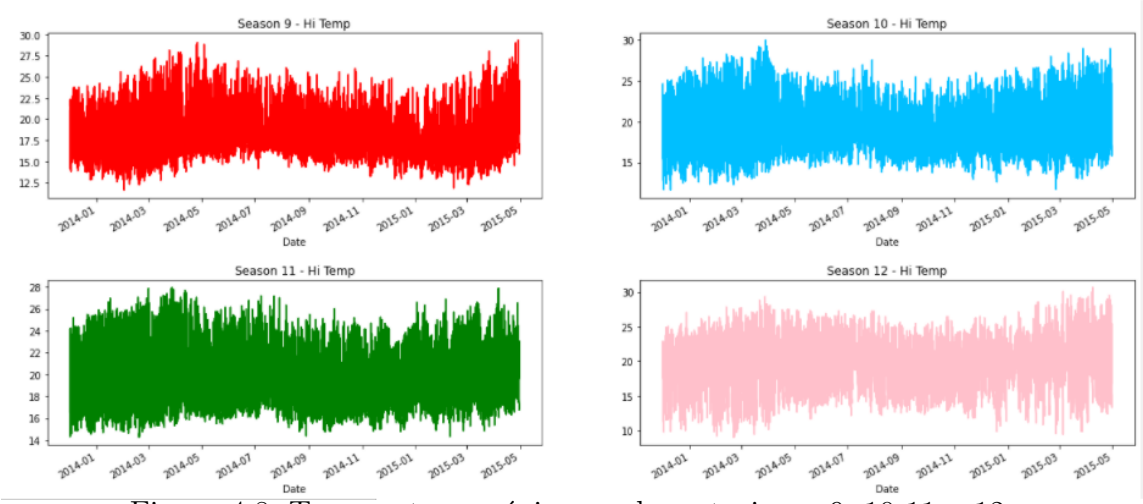


Figura 4.8: Temperatura máxima en las estaciones 9, 10 11 y 12.

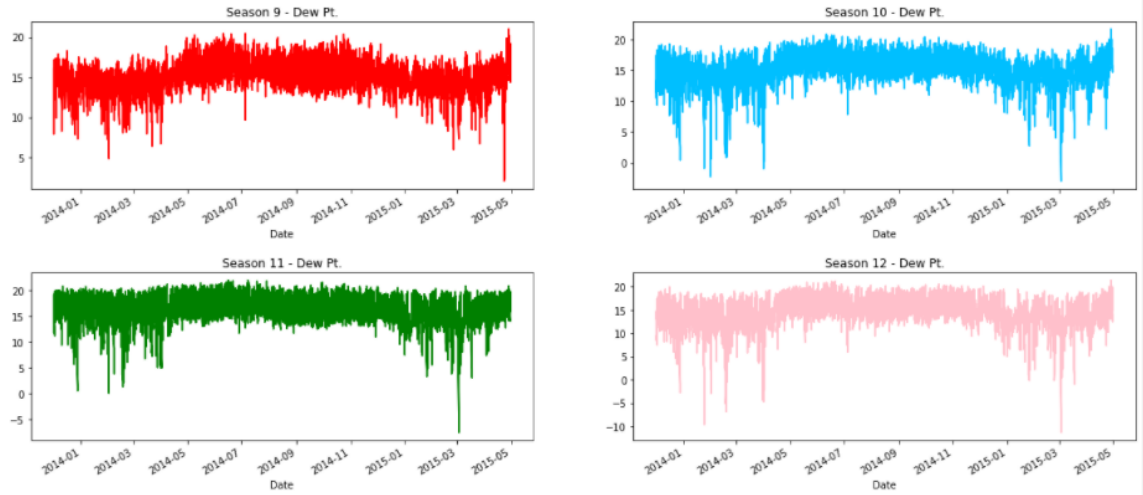


Figura 4.9: Punto de rocío en las estaciones 9, 10 11 y 12.

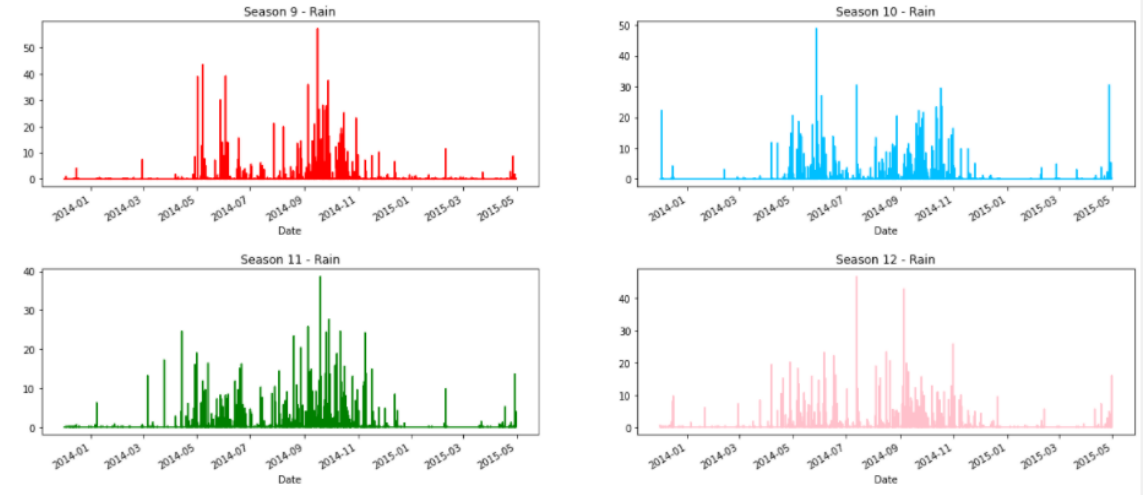


Figura 4.10: Precipitación en las estaciones 9, 10 11 y 12.

La gráfica 4.11 muestra las variables agrupadas en cada estación. Se puede visualizar que la humedad interior en la estación 9 y 11 muestran tendencias similares, y que la humedad interior en la estación 10 y 12 tienen comportamientos muy diferentes entre sí y en comparación a las estaciones 9 y 11. Se puede apreciar que para todas las estaciones las variables de tempertura máxima, velocidad del viento, precipitación y punto de rocío tienen tendencias similar entre ellas a través de todas las estaciones de la zona de Los Santos.

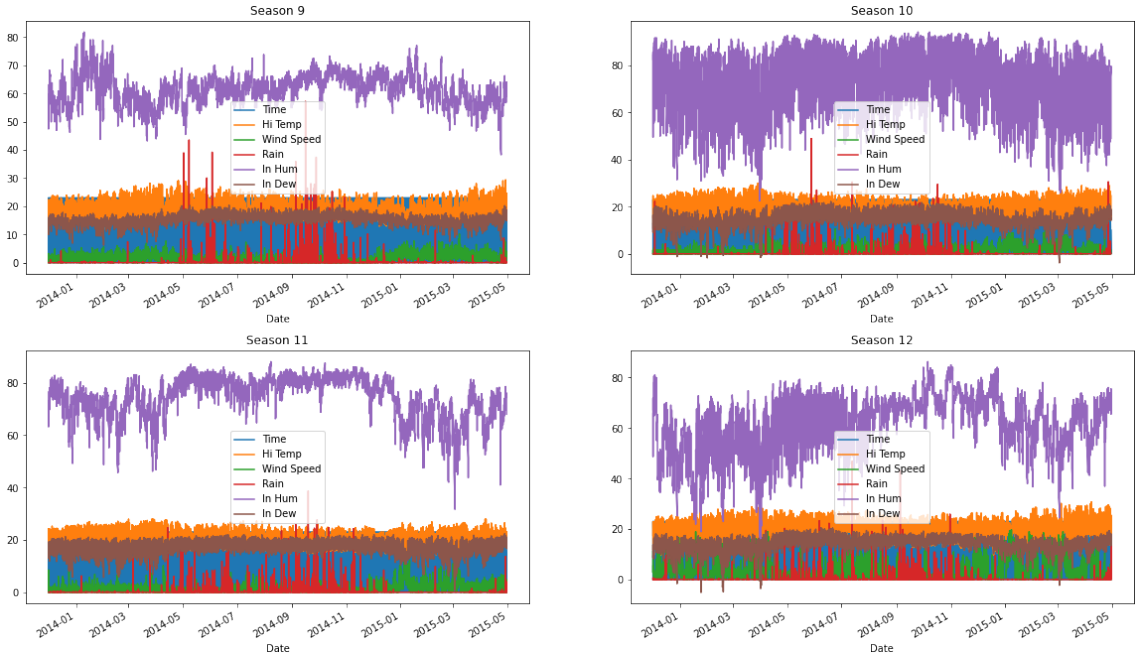


Figura 4.11: Variables no correlacionadas en las estaciones 9, 10 11 y 12.

En la gráfica 4.12 muestra el comportamiento de las variables seleccionadas para esta investigación y cómo se comportan en forma similar en las 4 estaciones seleccionadas anteriormente, siendo la temperatura exterior la variable que presenta un poco más de variación con respecto a las demás.

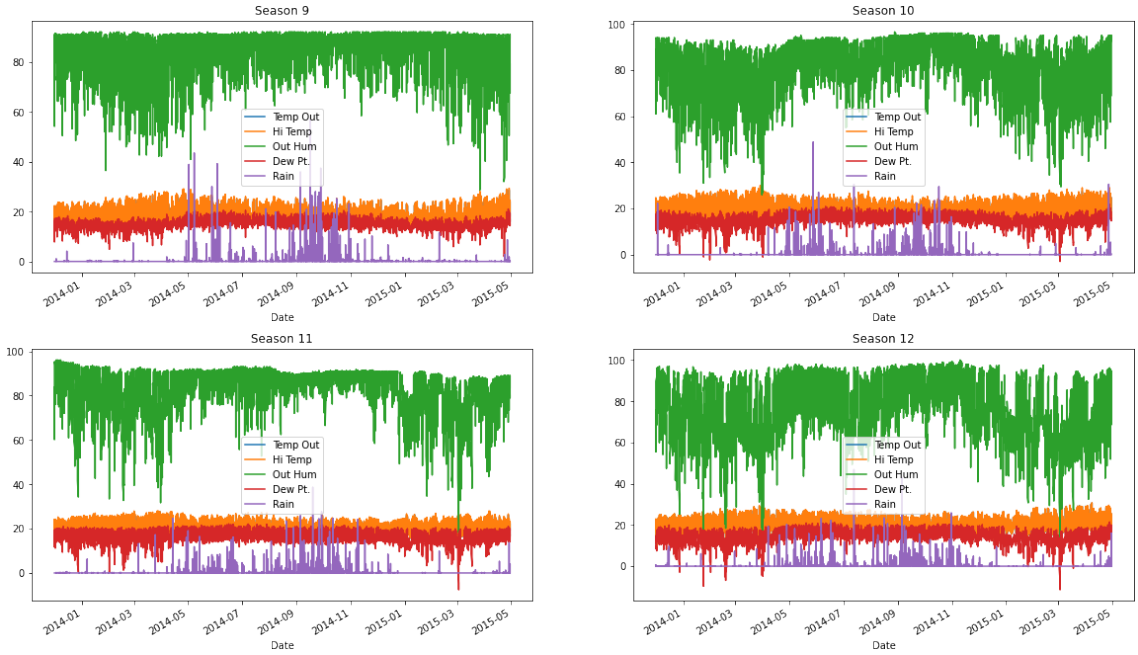


Figura 4.12: Variables en todas las estaciones 9, 10 11 y 12.

Las distribuciones multivariable deben tener una distribución normal por tres razones muy importantes:

1. La simplificación matemática: la distribución se torna más sencilla de trabajar.
2. Teorema del Límite Central en versión multivariable: si se tiene una colección de vectores aleatorios que son independientes e idénticamente distribuidos, entonces el vector medio muestral, será aproximadamente multivariante normalmente distribuido para grandes muestras.
3. Muchos fenómenos naturales pueden modelarse con distribuciones normales al igual que los casos univariados.

Se puede apreciar en las gráficas 4.13, 4.14, 4.15 y 4.16, cómo siguen una distribución normal o "casi"normal, el "casi"normal se debe principalmente a que los datos no están tan lejos de la normalidad dado que el "p-value" es muy pequeño en un conjunto de datos relativamente grande, lo que hace que algunos valores se desvíen de la "normalidad" [53, 54].

También se analizan, junto a las distribuciones, dos métricas importantes: la asimetría

"skewness" y el "curtosis". El "skewness" se carateriza por medir la asimetría de la distribución, este valor puede ser negativo o positivo y dependiendo del rango para este valor puede ser aceptada o rechazada la asimetría de normalidad. La medida del "curtosis" se caracteriza por medir si una distribución es de cola pesada o cola ligera en una distribución normal. El "curtosis" de una distribución normal es 3. Si una distribución determinada tiene una curtosis inferior a 3, se dice que es "playkurtic", lo que significa que tiende a producir menos valores atípicos y menos extremos que la distribución normal. Si una distribución determinada tiene una curtosis superior a 3, se dice que es "leptokurtic", lo que significa que tiende a producir más valores atípicos que la distribución normal [55].

Los valores de asimetría y curtosis entre -2 y +2 se consideran aceptables para demostrar una distribución univariada normal (George Mallery, 2010). Cabello et al. (2010) y Bryne (2010) argumentaron que los datos se consideran normales si la asimetría está entre -2 y +2 y la curtosis está entre -7 y +7 [56].

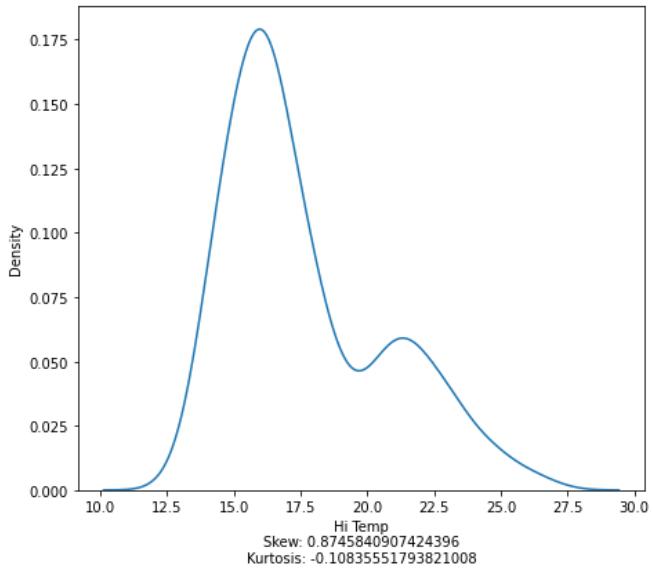


Figura 4.13: Distribución de los datos en la variable temperatura máxima estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal.

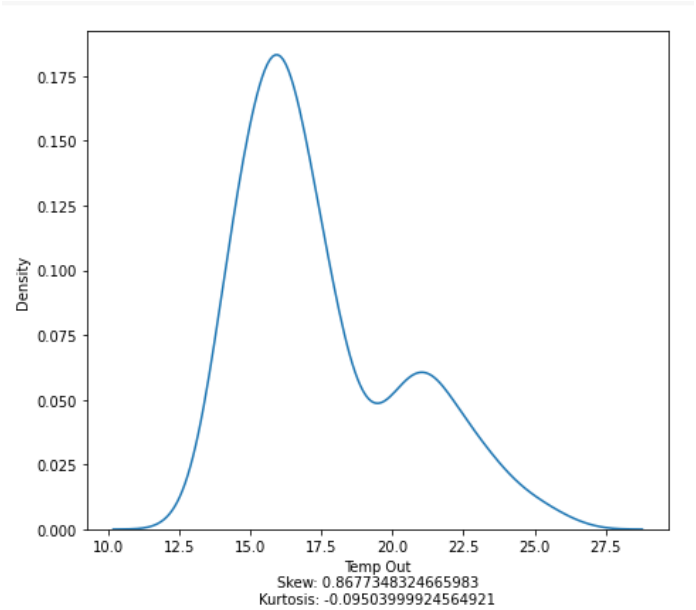


Figura 4.14: Distribución de los datos en la variable temperatura exterior estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal.

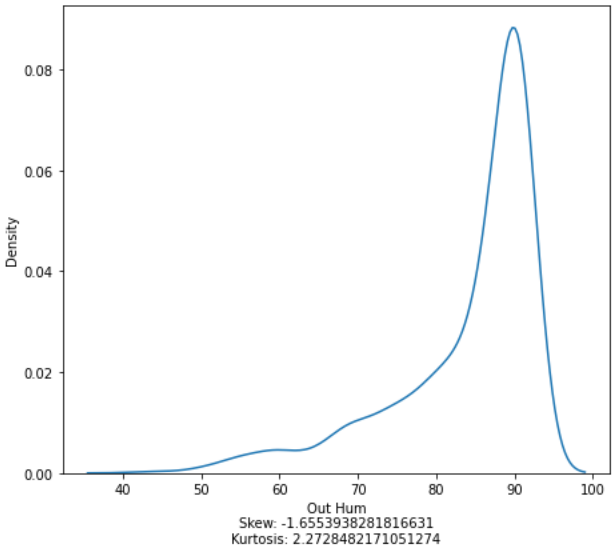


Figura 4.15: Distribución de los datos en la variable humedad exterior estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal.

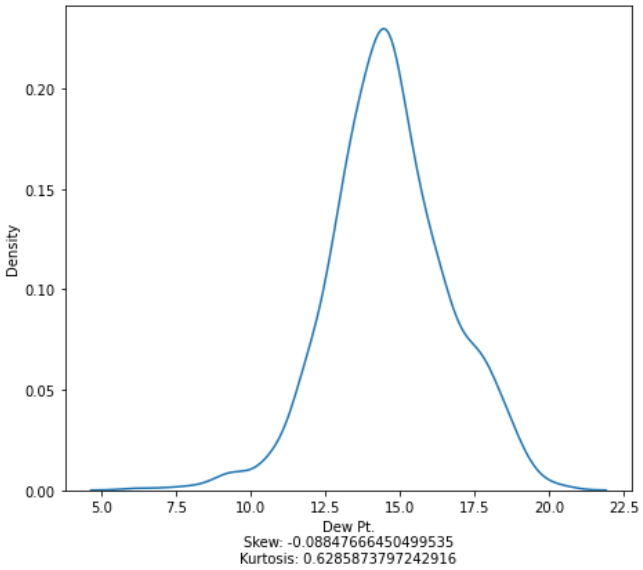


Figura 4.16: Distribución de los datos en la variable punto de rocío estación 9. Kurtosis y Skew cumplen las afirmaciones para conciderarse una distribución normal.

A diferencia de las demás variables, la precipitación tiene una una distribución similar a la distribución Pareto [57], pues el análisis de "skewness" y el "curtosis" en los rangos en que caen no pueden ser clasificadas de esta manera, sin embargo, según el análisis realizado por Boakye y Yao en [56], existen muchas maneras de determinar la normalidad en algunos casos. Por lo tanto, la variable de precipitación no cumple con el requerimiento de seguir una distribución normal y es una de las asunciones requeridas por el algoritmo EMB a la hora de imputar los datos, sin embargo en la sección de resultados se ampliará cómo las pruebas realizadas en EMB, con y sin precipitación, el algoritmo fue capaz de imputar los valores faltantes incluso cuando se trataba con la precipitación, sin elevar el error de forma importante.

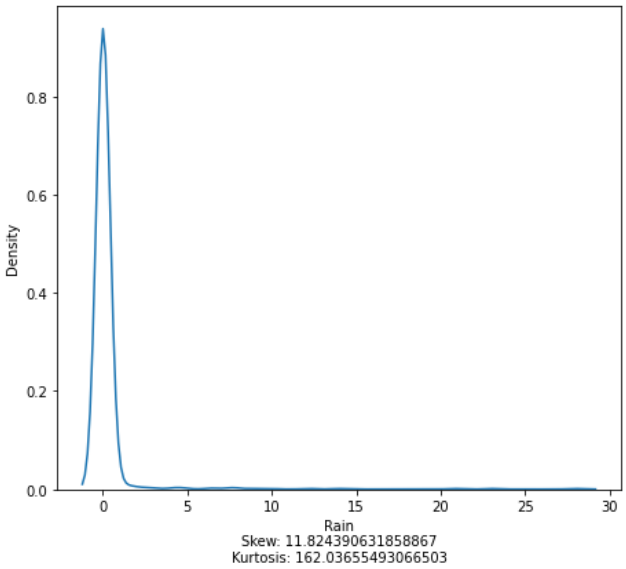


Figura 4.17: Distribución de los datos en la variable precipitación estación 9. Kurtosis y Skew no cumplen las afirmaciones para conciderarse una distribución normal.

4.2.4. Generación de datos faltantes para evaluación de imputaciones

La generación de archivos con porcentajes de datos faltantes para la ejecución del diseño experimental busca simular la problemática real experimentada por ICAFE cuando recopilan la información de los sensores. Varios escenarios pueden ocurrir durante la pérdida de datos. Por ejemplo, si los datos faltantes son durante la noche, es porque algunos sensores trabajan con paneles solares durante el día, pero en la noche entra a trabajar la batería, si la batería se agota o no son reemplazadas a tiempo, ocurre la pérdida de datos por falta de energía en el sensor. Si la pérdida de datos ocurre durante varios días consecutivos, el error puede ser debido a que la consola o datalogger al perder conectividad o electricidad se desconfiguran y dejan de almacenar la información (aunque el sensor esté enviando correctamente la información). Si la pérdida de datos se da solamente en algunas variables puede darse debido a que si son sensores de viento y no hay viento, el sensor no registra datos. También puede ocurrir un corto circuito (provocado por un animal, humedad o una gota de agua en el sensor) en el sensor que mide esa variable, por lo tanto, los datos no pueden ser enviados al datalogger.

Para la generación de los datos faltantes se procedió a generar archivos con faltantes aleatorios en las estaciones 9, 10 , 11 y 12 en las diferentes variables de modo que se tuvieran archivos con 1, 3, 5, 10 y 20 por ciento de datos faltantes. En este escenario, se busca evaluar los resultados de cada algoritmo a la hora de imputar datos faltantes en forma aleatoria (utilizado en el diseño experimental A, C, D).

Existen otros escenarios que se buscan evaluar en los diseños experimentales ampliados en este estudio. Por ejemplo, el diseño experimental B, busca someter a una prueba de estrés el rendimiento de los algoritmos utilizados para evaluar la capacidad de imputar datos faltantes en forma consecutiva (no mayor a 5 días consecutivos de datos faltantes), el diseño experimental E, que busca abordar de manera integral el caso de los datos faltantes y probar la efectividad de los algoritmos utilizados en la

imputación de datos faltantes (faltantes, por horas, faltantes por días, faltantes por variable y faltantes en todas las combinaciones) con las fallas frecuentes experimentadas en el ICAFE.

Arquitectura propuesta de aprendizaje automático para la imputación múltiple y multivariada de valores faltantes en series de tiempo del campo meteorológico utilizando LSTM

Para desarrollar la arquitectura propuesta de esta sección, es importante mencionar que mucha experimentación fue llevada a cabo a través de variaciones de arquitecturas LSTM, hiperparametrización con diferentes herramientas como Keras Tunner y Wandb para obtener una arquitectura que lograra mejoras con respecto a las fuentes utilizadas y consultadas en este estudio.

Los detalles de la experimentación llevada a cabo se mencionan en la sección del diseño experimental, sin embargo, en forma general, se puede mencionar que se utilizaron arquitecturas LSTM Encoder-Decoder, LSTM Encoder-Decoder Bidireccional, Transformer con LSTM Decoder, Redes Convolucionales, LSTM Bidireccionales, LSTM Encoder y Decoder Bidireccionales, LTSM Encoder con capas GRU, y otras

variantes de estas, con el fin de encontrar la mejor combinación de hiperparámetros y capas que nos brindara los mejores resultados en la imputación de datos faltantes en las series de tiempo.

La arquitectura de aprendizaje profundo desarrollada en esta investigación se basa en la arquitectura propuesta por Alhamid en [58]. En su trabajo, Alhamid propone una arquitectura a partir 2 capas ocultas, seguidas de 2 capas LSTM bidireccionales y una capa adicional oculta. La entrada es una secuencia de eventos y la salida es la predicción del siguiente evento en la secuencia. Tomando como base la arquitectura propuesta por Sucholutsky [36] se recopilan las recomendaciones y lineamientos brindados en su investigación de interés para la presente investigación: la utilización del optimizador Adam, y sugiere una RNN en forma bidireccional como trabajo futuro; utilizando LSTM bidireccional (considerando que esta sugerencia puede involucrar más latencia), utiliza 5 capas ocultas, sin embargo, concluye que capas adicionales no mejoran el rendimiento. Dadas las recomendaciones brindadas por los autores supra mencionados y a través de experimentación se logra desarrollar la siguiente arquitectura que mejora en un 2 % con respecto a las mejores arquitecturas LSTM evaluadas en la sección de diseño experimental A: 6.5, 6.7, 6.8, 6.9, 6.5.

A continuación, se describe la arquitectura utilizada en el presente estudio 5.1 que compite en cierta manera con algoritmos tan exitosos en la imputación de datos como lo son EMB y MissForest.

La capa de entrada de nuestro modelo es una capa Secuencial. Seguida de una capa LSTM con los siguientes parámetros: 64 unidades en la salida, inicializador kernel "he_uniform", inicializador recurrente "he_uniform", forma de la entrada: 5 pasos en el pasado, y 5 características a analizar, retorno de las secuencias en "True", y definiendo esta capa de entrada con el nombre de "enconder1".

La siguiente capa es una capa Bidireccional con parámetros merge mode con valor mul. Dicha capa Bidireccional envuelve una capa LSTM con parámetros: 32 unidades en la salida, inicializador kernel "he_uniform", inicializador recurrente "he_uniform", retorno de secuencias en "False" y finalmente un "dropout"de 0.5.

La siguiente capa es un Vector de Repetición con 5 pasos al futuro llamada “encoder_decoder_bridge”.

Seguida de otra capa Bidireccional que envuelve una capa LSTM con parámetros: 32 unidades en la salida, inicializador kernel “he_uniform”, inicializador recurrente “he_uniform”, retorno de secuencias en “True” y finalmente un “dropout”de 0.5, llamada “decoder 1”.

Nuevamente se agrega otra capa LSTM con parámetros: con 64 unidades en la salida, inicializador kernel “he_uniform”, inicializador recurrente “he_uniform”, retorno de secuencias en “True” llamada “decoder 2”.

La siguiente capa es un “GaussianDropout” con probabilidad de abandono o de caída (drop out) 0.5.

Finalmente la capa de salida es “Time Distributed”que contiene una capa Densa con parámetros: 5 características y con la función de activación “relu”.

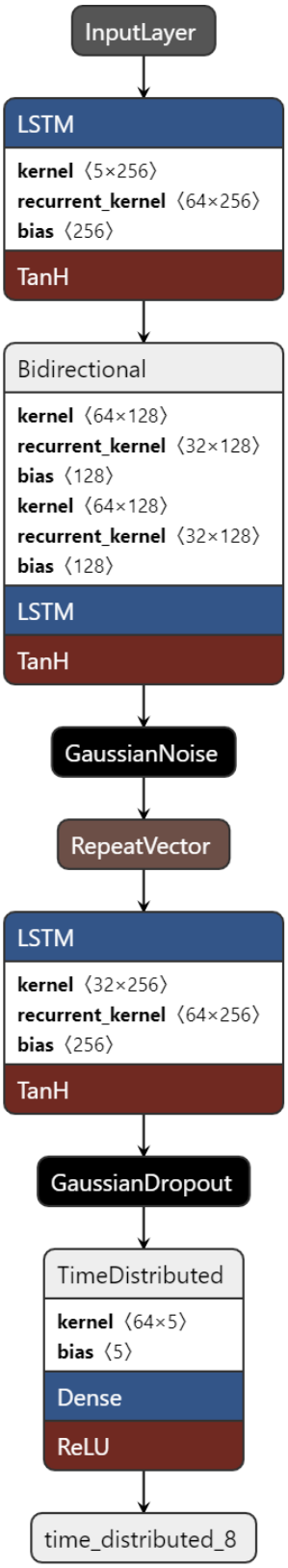


Figura 5.1: Arquitectura LSTM propuesta

5.0.1. Arquitectura propuesta de aprendizaje automático para la imputación múltiple y multivariada de valores faltantes en series de tiempo del campo meteorológico utilizando LSTM con conjuntos de datos correlacionados por variable según las estaciones aledañas

En esta sección se propone una variante a las pruebas ya realizadas, al obtener que el LSTM propuesto no mejora en forma significativa las imputaciones realizadas, por lo que se desea evaluar si las estaciones aledañas y las variables mejor correlacionadas pueden ayudar a mejorar las métricas obtenidas en los experimentos A, B y C.Se utiliza la misma arquitectura, pero se genera un nuevo connjunto de datos, de manera que si se pretende imputar la estación 10, en las variables de temperatura exterior, humedad exterior, temperatura máxima, precipitación y punto de rocío, para cada una de las variables se toman los valores de cada variable en las estaciones aledañas que guarden mejor grado de correlación y se construye un nuevo conjunto de datos tomando las columnas individuales predichas por los modelos que corresponden a las columnas que guardan mejor correlación con la estación que se pretende imputar. La gráfica 5.2, muestra cómo se construye el nuevo conjunto de datos de la estación 10, a partir de la búsqueda de las variables correlacionadas en las estaciones aledañas.

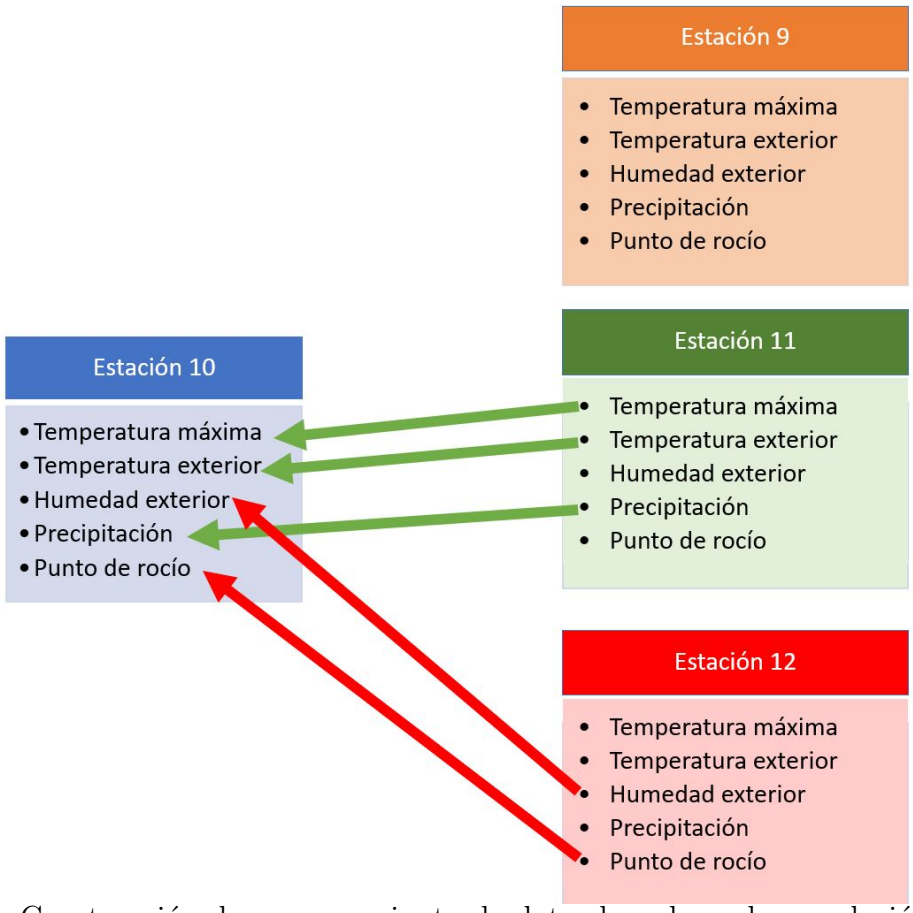
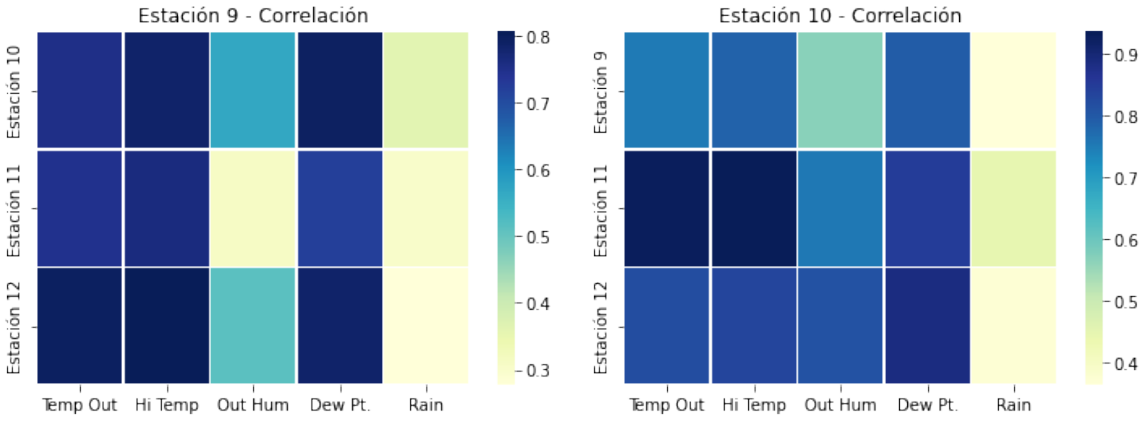


Figura 5.2: Construcción de nuevo conjunto de datos basado en la correlación entre las variables con las estaciones aledañas

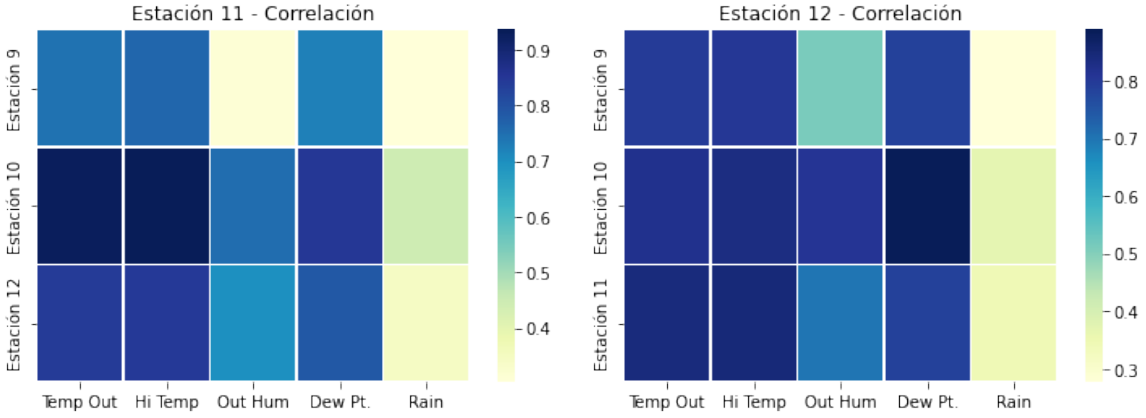
Se realizan los cálculos para generar una matriz de correlación que permita visualizar la correlación entre variables y estaciones de forma sencilla. Ver gráfica 5.3. De esta gráfica se pueden realizar dos deducciones muy importantes, la precipitación no guarda correlación con ninguna de las demás variables en cada estación. También se puede apreciar que la humedad exterior no tiene correlación en la estación 9 y la 11, ni tampoco en la estación 11 con la estación 9 en la misma variable.

Una vez generado el nuevo conjunto de datos, se procede a entrenarlo para luego proceder a imputar la estación deseada, utilizando cada característica que guarda mejor correlación. Para esta prueba se genera un diseño experimental D, en dónde se evalúan los resultados de realizar esta prueba.



(a) Matriz de correlación, estación 9

(b) Matriz de correlación, estación 10



(c) Matriz de correlación, estación 11

(d) Matriz de correlación, estación 12

Figura 5.3: Matriz de correlación entre estaciones y variables

5.0.2. Hiperparametrización

En esta sección, se detallan los experimentos ejecutados y se analizan los resultados obtenidos. Los experimentos se dividen en dos fases: hiperparametrización y experimentos finales. En la fase de hiperparametrización se eligen los mejores hiperparámetros de cada modelo a analizar, para luego utilizarlos en la fase de experimentos finales. Los experimentos finales son el resultado de la depuración de la sección de hiperparametrización de los modelos.

Hiperparametrización del modelo EMB

El algoritmo EMB no requirió ninguna hiperparametrización especial (pues al tratar de hiperparametrizarlo no aportaba mejoras al modelo y agregaba tiempo de ejecución), se imputaron los datos faltantes con la configuración "default"[59]. El algoritmo está representado en su forma más básica al no especificar columna de tiempo, pues esto genera más tiempo de procesamiento, por tanto, la complejidad temporal no está limitada por algún polinomio (el parámetro "polytime" indica el poder del polinomio a ser incluido en el modelo de imputación).

Hiperparametrización del modelo Random Forest: Missforest

El algoritmo MissForest no fue hiperparametrizado (a excepción del diseño experimental B, sin embargo de igual manera ni la configuración "default"ni ninguna otra variante logró afrontar el problema de rellenar grandes "huecos"de datos), ya que agregar o restar árboles de decisión, que es lo que puede hiperparametrizarse, no aportaba mejoras al modelo de predicción, los demás valores que se podrían hiperparametrizar aplican bien a este caso, por tanto no fueron modificados. Sin embargo, al agregar más árboles de decisón no se presentaron mejoras, por lo tanto se utilizó la configuración "default". Dentro de las posibilidades que presenta el algoritmo, se puede realizar el "fit" con una estación y ejecutar el "transform" con otra estación, a diferencia de EMB que solamente puede imputar sobre una estación sin realizar algún entrenamiento o análisis previo.

Hiperparametrización del modelo LSTM propuesto

El modelo LSTM múltiple y multivariado desarrollado en esta investigación, fue hiperparametrizado utilizando la metodología de búsqueda aleatoria sobre los parámetros: timesteps, epochs, learning rate, merge mode y loss. En donde, basados en la investigación Sucholutsky [36] y la exploración de parámetros realizada en esta investigación, se muestran las mejores combinaciones de los parámetros que brindan las mejores

métricas para la imputación de los datos en los conjuntos de datos mencionados anteriormente con el fin de utilizarlos en los experimentos finales. El cuadro 5.1 muestra los hiperparámetros y los valores que dieron mejores resultados en la hiperparametrización del presente modelo. En otras palabras, se escogió el modelo que proporciona el mejor rendimiento con el objetivo de maximizar la métrica RMSE. En total, todos los experimentos abarcan 7 variables diferentes, con 800 combinaciones de valores distintos. El cuadro 5.1 muestra las variables probadas durante la hiperparametrización. Para reducir la cantidad de experimentos, se utilizaron siempre los conjuntos estación 9 y estación 10. Para elegir los mejores resultados en las imputaciones de datos se tomaron en cuenta las métricas arrojadas por RMSE, MAE y R2 en los algoritmos (detallados en la siguiente sección) a la hora de predecir los datos imputados para el 1 %, 3 %, 5 %, 10 % y 20 % de los datos faltantes.

Hiperparámetro	**Valores Probados**
batch size	128, 64, **32**, 16
activation function	**relu**, sigmoid
timesteps	72, 48, **24**, 12, 5
epochs	10, 20, 50, 80, **100**
learning rate	**0.0001**
merge mode	sum, mul, **concat**, ave
loss	**mse**

Cuadro 5.1: Hiperparámetros y valores utilizados en la búsqueda aleatoria del modelo LSTM múltiple y multivariado.

Experimentos finales y análisis de resultados

Los experimentos finales comprenden los 15 experimentos establecidos en la sección 4.1.4 y detallados en la sección Diseño experimental A (detallada más adelante). Pero además, durante la experimentación se encontraron nuevas opciones que se consideraron importantes de evaluar e incluir en los resultados producto de esta investigación. Es importante aclarar que la imputación de variables se realiza sobre cualquiera de las cinco variables utilizadas en este estudio (precipitación, punto de rocío, temperatura máxima, humedad exterior, y temperatura exterior).

Los resultados de la hiperparametrización de la arquitectura LSTM propuesta apuntaron a que los mejores valores durante la hiperparametrización son los especificados en la tabla 6.1.

Hiperparámetro	**Valor seleccionado**
batch size	32
activation function	relu
timesteps	24
epochs	100
learning rate	0.0001
merge mode	concat
loss	mse

Cuadro 6.1: Hiperparámetros y valores utilizados en la búsqueda aleatoria del modelo LSTM múltiple y multivariado.

Para todos los experimentos planteados en los diseños experimentales, se evalúan las métricas MAE, RMSE , R2 y el tiempo de ejecución, siendo el RMSE la principal métrica para determinar la efectividad de las imputaciones en los diferentes algoritmos. Para ello, esta sección se dividió en cuatro diseños experiementales detallados a continuación.

Diseño experimental A (original): Se utiliza el conjunto de datos de la estación 9 para entrenamiento de la arquitectura LSTM propuesta. Y para la validación y pruebas se utilizan los conjuntos de datos de las estaciones 10, 11 y 12. Con datos comprendidos entre diciembre 2013 y abril 2015, y con valores borrados al azar al 1 %, 3 %, 5 %, 10 % y 20 %, dando como resultado un total de 15 experimentos. En tabla 6.2 se presentan los resultados obtenidos, se puede observar que el MAE, RMSE y R2 tienen valores similares entre las estaciones entre sí para cada uno de los porcentajes de valores faltantes.

También se comparan los algoritmos EMB y MissForest con la arquitectura LSTM propuesta. Como limitaciones, se encuentra que, EMB solamente puede imputar datos utilizando la misma estación (se utiliza la estación 10), MissForest puede hacer fit con una estación y hacer imputación de otra estación (por lo que se hace fit con la estación 9 e imputación en la estación 10), generando 10 experimentos más y comparando con el experimento generado con la arquitectura LSTM propuesta con la estación 10. En la tabla 6.3 y en la gráfica 6.1 se comparan los algoritmos EMB, MissForest y el LSTM propuesto, entrenando e imputando los mismos datos para verificar la eficacia de estos algoritmos a la hora de imputar y así poder compararlos bajo los mismos criterios. Los resultados basados en el análisis del RMSE en la imputación en forma general (para la imputación de todos las variables) muestran que MissForest y EMB lo hacen mucho mejor que la arquitectura LSTM propuesta. En la gráfica 6.1, se puede apreciar que la arquitectura LSTM propuesta obtiene el valor del RMSE más grande (en comparación con los demás algoritmos) para cada uno de los porcentajes de datos faltantes evaluados, indicando mayor error a la hora de

imputar, sin embargo, para el 1 %, 3 %, 5 % y 10 % de datos faltantes en el R2 la arquitectura LSTM compite con métricas similares contra MissForest y el algoritmo EMB. Las gráficas 6.1, 6.2 y 6.3 pretenden mostrar visualmente el comportamiento de las métricas analizadas con las imputaciones de datos en los diferentes porcentajes de datos faltantes.

Para una mejor referencia del comportamiento, se puede apreciar la media y las desviaciones estándar en la tabla 6.3

Adicionalmente, se incluyen en esta sección otros experimentos realizados con otras arquitecturas LSTM con el fin de buscar las mejores variantes en LSTM en series de tiempo: LSTM + Loung Attention, LSTM + Red Convolucional, Tranformer + LSTM, LSTM Encoder-Decoder y LSTM Bidirectional, dando como resultado 21 experimentos adicionales, pero no se toman en cuenta para definir los resultados de este diseño experimental, sino que pretenden mostrar el trabajo realizado en analizar otras variantes LSTM y su efectividad en imputación de datos de series de tiempo, donde la arquitectura LSTM propuesta en este estudio muestra una mejora leve con respecto a estas arquitecturas. El resultado de estas pruebas muestran que otros LSTM evaluados fueron inferiores a los obtenidos con la arquitectura LSTM propuesta en la imputación de datos, 21 experimentos adicionales son detallados en las gráficas 6.5, 6.6, 6.7, 6.8, 6.9 en donde ninguna de las arquitecturas evaluadas presenta una mejora que supere el 80 % en el R2 y que el RMSE sea inferior a 2.62 para el 20 % de datos faltantes en comparación a LSTM propuesto que obtiene un 82 % en el R2 y el RMSE es menor a 2.47 para el 20 % de datos faltantes.

	E10			E11			E12		
	MAE	**RMSE**	**R2**	**MAE**	**RMSE**	**R2**	**MAE**	**RMSE**	**R2**
1 %	0.02	0.54	0.99	0.02	0.51	0.99	0.032	0.61	0.99
3 %	0.07	0.91	0.98	0.06	0.84	0.99	0.10	1.03	0.98
5 %	0.12	1.20	0.97	0.10	1.03	0.98	0.16	1.31	0.97
10 %	0.26	1.74	0.91	0.21	1.45	0.94	0.34	1.91	0.94
20 %	0.58	2.47	0.82	0.48	2.06	0.84	0.77	2.99	0.83

Cuadro 6.2: Diseño experimental A. Métricas resultantes de la hiperparametrización de la arquitectura LSTM propuesta en la imputación de datos sobre las estaciónes 10, 11, 12 entrenadas con la estación 9. Comprendiendo 15 experimentos en total en esta sección

	EMB						MissForest						LSTM propuesto					
	MAE	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**
1 %	0.01	0.11	1.00	1.86s	26.76	29.05	0.00	0.06	1.00	56s	26.75	29.05	0.02	0.54	0.99	4s	26.85	29.15
3 %	0.03	0.32	0.99	2.48s	26.76	29.05	0.01	0.24	1.00	55s	26.76	29.05	0.07	0.91	0.98	4s	26.80	29.09
5 %	0.05	0.42	0.99	1.72s	26.77	29.04	0.016	0.30	0.99	1:05m	26.76	29.05	0.12	1.20	0.97	5s	26.82	29.11
10 %	0.12	0.88	0.93	2.08s	26.78	29.03	0.05	0.65	0.96	37s	26.76	29.05	0.26	1.74	0.91	5s	26.77	29.07
20 %	0.32	1.70	0.88	2.84s	26.80	29.04	0.17	1.24	0.92	52s	26.77	29.05	0.58	2.47	0.82	4s	26.80	29.07

Cuadro 6.3: Diseño experimental A. Métricas resultantes en la imputación de datos sobre la estación 10 con el algoritmo EMB, MissForest y el algoritmo LSTM propuesto

	Datos Originales	
	Mean	**STDev**
Estación 09	26.84	29.78
Estación 10	28.90	32.09
Estación 11	29.46	32.02
Estación 12	27.84	30.86

Cuadro 6.4: Métricas de media y desviación estándar sobre los datos originales en la estación 10 con el algoritmo EMB, MissForest y el algoritmo LSTM propuesto

	% Missing	Model	MAE	RMSE	R2
0	1%	MissForest	0.001964	0.055620	0.999374
1	1%	EMB	0.007387	0.114388	0.998293
2	1%	LSTM-BD Gaussian Noise-Dropout	0.023446	0.535071	0.995799
4	3%	MissForest	0.008899	0.235099	0.995797
5	3%	EMB	0.025735	0.318185	0.992321
6	3%	LSTM-BD Gaussian Noise-Dropout	0.072199	0.908653	0.984902
8	5%	MissForest	0.016534	0.296101	0.991515
9	5%	EMB	0.045640	0.418722	0.983573
10	5%	LSTM-BD Gaussian Noise-Dropout	0.119106	1.194878	0.973241
12	10%	MissForest	0.052144	0.646780	0.960178
13	10%	EMB	0.116979	0.882636	0.933875
14	10%	LSTM-BD Gaussian Noise-Dropout	0.259043	1.737752	0.912823
16	20%	MissForest	0.172935	1.243434	0.920088
17	20%	EMB	0.306823	1.616151	0.881047
18	20%	LSTM-BD Gaussian Noise-Dropout	0.578064	2.469557	0.818461

Figura 6.1: Métricas obtenidas en la comparación de los algoritmos MissForest, EMB y el LSTM propuesto, imputando con los diferentes porcentajes valores faltantes.

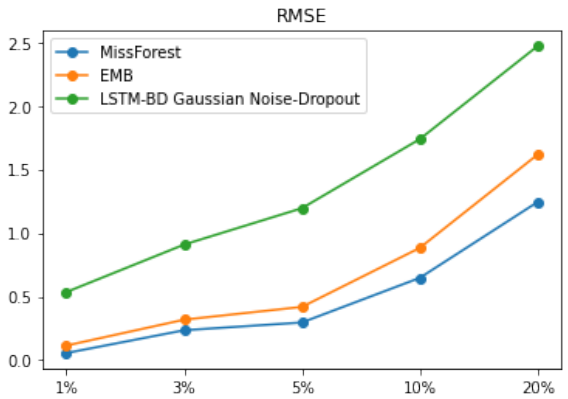


Figura 6.2: Gráfica que analiza el RMSE de los algoritmos MissForest, EMB y LSTM propuesto en los diferentes porcentajes de datos faltantes

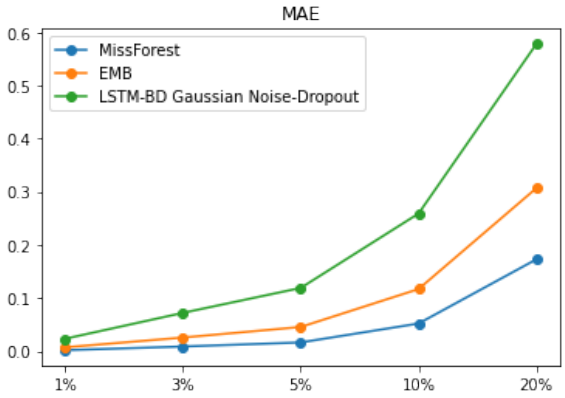


Figura 6.3: Gráfica que analiza el MAE de los algoritmos MissForest, EMB y LSTM propuesto en los diferentes porcentajes de datos faltantes

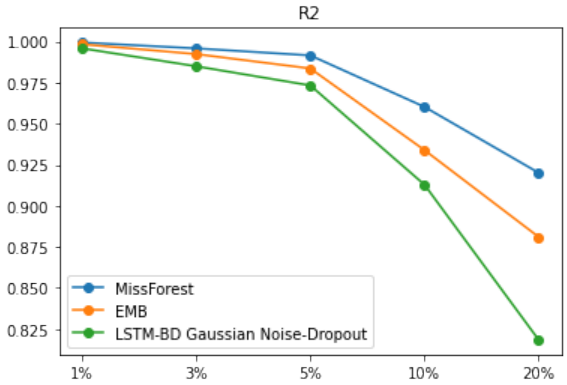


Figura 6.4: Gráfica que analiza el R2 de los algoritmos MissForest, EMB y LSTM propuesto en los diferentes porcentajes de datos faltantes

	RMSE	**R2**	**Duración**
1 %	77.23	-639.51	NA

Cuadro 6.5: Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el algoritmo LSTM + Loung, Attention, muchos valores en negativo, no aprende, se descarta esta arquitectura

	RMSE	**R2**	**Duración**
1 %	0.59	0.99	1 seg
3 %	1.02	0.97	1 seg
5 %	1.33	0.95	1 seg
10 %	1.91	0.88	1 seg
20 %	2.69	0.76	1 seg

Cuadro 6.6: Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo Red convolutional + LSTM

	RMSE	R2	Duración
1 %	0.56	0.99	1 seg
3 %	0.99	0.98	1 seg
5 %	1.31	0.96	1 seg
10 %	1.93	0.88	1 seg
20 %	2.68	0.77	1 seg

Cuadro 6.7: Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo Transformer + LSTM

	RMSE	R2	Duración
1 %	0.54	0.99	4 seg
3 %	0.92	0.98	4 seg
5 %	1.22	0.97	4 seg
10 %	1.80	0.90	4 seg
20 %	2.62	0.80	4 seg

Cuadro 6.8: Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo LSTM Encoder-Decoder

	RMSE	R2	Duración
1 %	0.54	0.99	3 seg
3 %	0.92	0.98	3 seg
5 %	1.23	0.97	3 seg
10 %	1.83	0.91	3 seg
20 %	2.76	0.80	3 seg

Cuadro 6.9: Métricas resultantes en la imputación de datos sobre la estación 10 y haciendo entrenamiento con la estación 9, con el modelo LSTM Bidireccional

Diseño experimental B: Esta variante contempla la posibilidad de que los datos faltantes se presenten en forma secuencial para todas las variables, es decir, el sensor que registra los datos del clima pierde por completo el rastreo de las variables por un periodo de 2 a 5 días consecutivos hasta completar el 1 %, 3 %, 5 %, 10 % y 20 % de datos faltantes. Se pretendía evaluar los mismos 15 experimentos planteados en el Diseño Experimental A, pero con datos faltantes en forma consecutiva, se observó que: el algoritmo MissForest no logra manejar la imputación de datos consecutivos, el algoritmo EMB a pesar que tiene como supuesto para el rendimiento del algoritmo que los datos faltantes deben encontrarse al azar, logra imputar con mejor efectividad los datos faltantes respecto a los otros dos algoritmos y finalmente el algoritmo LSTM propuesto no logra imputar, solamente repite datos de las ventanas previas, no tiene información de otras variables (solamente de los pasos pasados) y, por lo tanto, la imputación de datos no es buena, se descarta seguir evaluando este experimento con datos faltantes mayores a 1 %.

Diseño experimental C: Para esta propuesta de diseño experimental, al evaluar la arquitetura LSTM propuesta en forma general y no presentar los resultados esperados, y, además comprendiendo que las redes neuronales recurrentes requieren de muchos datos y en los experimentos anteriores la cantidad de datos presente por estación es relativamente poca (1 año y 4 meses), se procede a evaluar otros rangos (en años) de las estaciones 8 y 9 con datos presentes desde 2014 al 2018 (duplicando la cantidad de datos). Se vuelven a evaluar los algoritmos evaluados en los diseños A y B, pero con mayor cantidad de datos y los archivos de datos faltantes con datos eliminados al azar, con 1 %, 3 %, 5 %, 10 % y 20 %, dando como resultado otros 15 experimentos.

	EMB						MissForest						LSTM propuesto					
	MAE	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**
1 %	0.01	0.10	1.00	4.66s	27.27	29.64	0.00	0.07	1.00	2.24m	27.27	29.65	0.02	0.52	0.99	12s	27.28	29.66
3 %	0.02	0.21	0.99	4.9s	27.27	29.64	0.00	0.15	0.99	2.22m	27.26	29.65	0.06	0.79	0.98	12s	27.29?	29.67
5 %	0.03	0.30	0.98	5.00s	27.27	29.64	0.01	0.20	0.99	1.54m	27.26	29.64	0.10	1.03	0.96	12s	27.31	29.68
10 %	0.07	0.55	0.96	5.17s	27.28	29.63	0.03	0.36	0.98	3.18m	27.27	29.65	0.22	1.43	0.92	23s	27.34	29.70
20 %	0.22	1.21	0.89	5.43s	27.30	29.62	0.11	0.88	0.93	1.51m	27.26	29.63	0.50	2.04	0.79	12s	27.27	29.63

Cuadro 6.10: Diseño Experimental C. Métricas resultantes en la imputación de datos sobre la estación 9 con el algoritmo EMB, MissForest y LSTM propuesto. Las métricas resultantes haciendo entrenamiento con la estación 8 e imputando sobre la estación 9.

	EMB						MissForest						LSTM propuesto					
	MAE	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**
1 %	0.01	0.11	1.00	1.86s	26.76	29.05	0.00	0.06	1.00	56s	26.75	29.05	0.02	0.54	0.99	4s	26.85	29.15
3 %	0.03	0.32	0.99	2.48s	26.76	29.05	0.01	0.24	1.00	55s	26.76	29.05	0.07	0.91	0.98	4s	26.80	29.09
5 %	0.05	0.42	0.99	1.72s	26.77	29.04	0.016	0.30	0.99	1:05m	26.76	29.05	0.12	1.20	0.97	5s	26.82	29.11
10 %	0.12	0.88	0.93	2.08s	26.78	29.03	0.05	0.65	0.96	37s	26.76	29.05	0.26	1.74	0.91	5s	26.77	29.07
20 %	0.32	1.70	0.88	2.84s	26.80	29.04	0.17	1.24	0.92	52s	26.77	29.05	0.58	2.47	0.82	4s	26.80	29.07

Cuadro 6.11: Diseño Experimental A. Copia de la gráfica 6.2 para utilizar como referencia para comparar con la tabla anterior el diseño experimental A y el diseño experimental C.

Para una mejor referencia del comportamiento, se puede apreciar la media y las

desviaciones estándar en la tabla 6.10

En general, se puede sintetizar que mayor cantidad de datos requirió más tiempo de entrenamiento e imputación durante el análisis del tiempo de ejecución. Las métricas son muy similares a cuando se hizo con los datos del experimento A (menos de la mitad de datos) (ver gráfica 6.14 y comparar con la gráfica 6.10). Se puede apreciar que, a mayor cantidad de datos, hay mejoras en las métricas MAE, RMSE y R2 en la evaluación del LSTM propuesto.

Diseño experimental D: En este diseño experimental, se plantea evaluar los modelos utilizados previamente, pero creando un nuevo conjunto de datos, utilizando cada variable de la estación mejor correlacionada con la variable en cuestión. De esta manera, se crea un nuevo conjunto de datos mixto que contiene las 5 variables tomadas de las estaciones mejor correlacionadas. Para ilustrar lo anterior, si se está imputando la estación 9, en la variable de precipitación, se busca cuál de las estaciones cercanas 10, 11 o 12 tiene mejor correlación en la variable de precipitación y se toma toda esa columna de datos, de forma sucesiva, se hace el mismo procedimiento con las variables restantes. Se genera un modelo entrenado sobre la estación mejor correlacionada a la actual, siendo la actual estación sobre la cual se pretende hacer el rellenado de datos faltantes. Los datos faltantes a imputar se evalúan en forma aleatoria con 1 %, 3 %, 5 %, 10 % y 20 % de datos faltantes, dando como resultado 15 experimentos adicionales para la nueva evaluación de la arquitectura.

	E10			**E11**			**E12**		
	MAE	**RMSE**	**R2**	**MAE**	**RMSE**	**R2**	**MAE**	**RMSE**	**R2**
1 %	0.02	0.54	0.99	0.02	0.51	0.99	0.03	0.62	0.99
3 %	0.07	0.91	0.99	0.06	0.86	0.98	0.10	1.03	0.98
5 %	0.12	1.20	0.97	0.10	1.05	0.97	0.16	1.32	0.97
10 %	0.27	1.77	0.91	0.23	1.50	0.93	0.34	1.92	0.93
20 %	0.62	2.57	0.78	0.51	2.23	0.83	0.77	2.95	0.82

Cuadro 6.12: Imputación de datos sobre las estaciones 10, 11, 12 entrenadas con la estación 9. Experimento D

	EMB						MissForest						LSTM propuesto					
	MAE	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**
1 %	0.01	0.11	1.00	1.86s	26.76	29.05	0.00	0.06	1.00	1.06m	26.75	29.05	0.02	0.54	0.99	12s	26.77	29.07
3 %	0.03	0.31	0.99	2.48s	26.76	29.05	0.01	0.24	0.99	48s	26.76	29.05	0.07	0.91	0.99	8s	28.95	32.13
5 %	0.05	0.42	0.98	1.72s	26.77	29.04	0.02	0.30	0.99	47s	26.76	29.05	0.12	1.20	0.97	8s	26.82	32.17
10 %	0.12	0.88	0.93	2.08 s	26.78	29.03	0.05	0.65	0.96	46s	26.78	29.03	0.27	1.77	0.91	9s	26.85	32.22
20 %	0.31	1.69	0.88	2.84 s	26.80	29.04	0.17	1.26	0.92	53s	26.77	29.05	0.62	2.57	0.78	8s	26.83	32.21

Cuadro 6.13: Diseño experimental D. Métricas resultantes en la imputación de datos sobre la estación 10 con la arquitectura LSTM (EMB, MissForest utilizan las métricas obtenidas en el experimento A y se utilizan como referencia para comparación).

	EMB						MissForest						LSTM propuesto					
	MAE	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**	**MAE**	**RMSE**	**R2**	**Dur**	**Mean**	**STDev**
1 %	0.01	0.11	1.00	1.86s	26.76	29.05	0.00	0.06	1.00	56s	26.75	29.05	0.02	0.54	0.99	4s	26.85	29.15
3 %	0.03	0.32	0.99	2.48s	26.76	29.05	0.01	0.24	1.00	55s	26.76	29.05	0.07	0.91	0.98	4s	26.80	29.09
5 %	0.05	0.42	0.99	1.72s	26.77	29.04	0.016	0.30	0.99	1:05m	26.76	29.05	0.12	1.20	0.97	5s	26.82	29.11
10 %	0.12	0.88	0.93	2.08s	26.78	29.03	0.05	0.65	0.96	37s	26.76	29.05	0.26	1.74	0.91	5s	26.77	29.07
20 %	0.32	1.70	0.88	2.84s	26.80	29.04	0.17	1.24	0.92	52s	26.77	29.05	0.58	2.47	0.82	4s	26.80	29.07

Cuadro 6.14: Diseño Experimental A. Copia de la gráfica 6.2 para utilizar como referencia para comparar con la tabla anterior el diseño experimental A y el diseño experimental D.

Para una mejor referencia del comportamiento, se puede apreciar la media y las

desviaciones estándar en la tabla 6.13

Para evaluar las imputaciones realizadas (ver 6.12) con LSTM en las estaciones 10, 11 y 12, se puede apreciar que en la estación 10 para el 10 % y 20 % de los datos faltantes, la imputación presenta una desmejora con respecto a las estaciones 11 y 12 y con respecto a las métricas obtenidas en el diseño experimental A (ver 6.2).

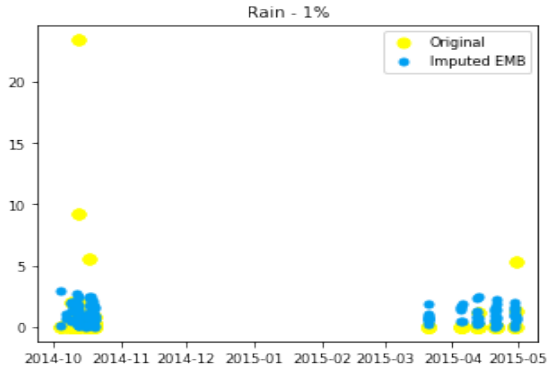
Al comparar los experimentos A (ver 6.2) y D (ver 6.13) se concluye que, a pesar de que se busca imputar las estaciones por característica en correlación con otras estaciones de la misma zona geográfica, al combinar las variables de las estaciones para generar un modelo a entrenar, las métricas se desestabilizan y esto podría explicarse por la pérdida de estacionalidad y tendencia entre las variables. El experimento D no presenta mejoras importantes en las métricas de imputación con LSTM en relación con el experimento A.

Diseño experimental E: En este punto se ha evaluado datos faltantes aleatorios, datos faltantes consecutivos produciendo grandes "huecos"de datos faltantes y, en esta prueba de diseño experimental, se pretende simular los escenarios de casos reales a los que se expone el ICAFE cuando hay pérdida de datos, especificados en la sección 4.2.4 (falta de batería en el sensor solar, faltantes en horas aleatorias por corto circuito, faltantes en rangos menores a las 24 horas debido a la desincronización con el datalogger). Es importante aclarar que los datos faltantes desarrollados en esta sección fueron generados en forma manual simulando los posibles escenarios de datos faltantes en ICAFE. Se añaden 9 diseños experimentales más, imputando la estación 10 y se pretende comparar las métricas obtenidas. A su vez, también se busca proporcionar una visualización enfocada a las imputaciones realizadas por cada algoritmo y compararlas con respecto a los datos reales. A continuación, en la tabla 6.15, se muestran las métricas obtenidas en este diseño y se puede deducir que la arquitectura LSTM propuesta, se desempeña mejor que los algoritmos de imputación EMB y MissForest.

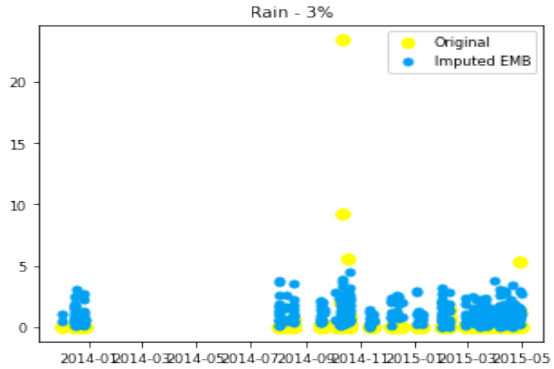
	EMB						MissForest						LSTM propuesto					
	MAE	**RMSE**	**R2**	**D**	**Mean**	**STDv**	**MAE**	**RMSE**	**R2**	**D**	**Mean**	**STDv**	**MAE**	**RMSE**	**R2**	**D**	**Mean**	**STDv**
1 %	0.05	0.78	0.98	2.18s	26.74	29.02	0.03	0.43	0.99	1.16m	26.75	29.05	0.02	0.40	0.99	5s	26.75	29.04
3 %	0.14	1.32	0.94	2.35s	26.74	29.00	0.09	1.00	0.97	40s	26.76	29.07	0.08	0.93	0.98	5s	26.77	29.06
5 %	0.23	1.68	0.92	2.18s	26.75	29.00	0.15	1.32	0.95	40s	26.78	29.08	0.13	1.25	0.96	5s	26.80	29.09

Cuadro 6.15: Diseño Experimental E. Métricas obtenidas al evaluar EMB, MissForest y LSTM propuesto en la imputación de la estación 10.

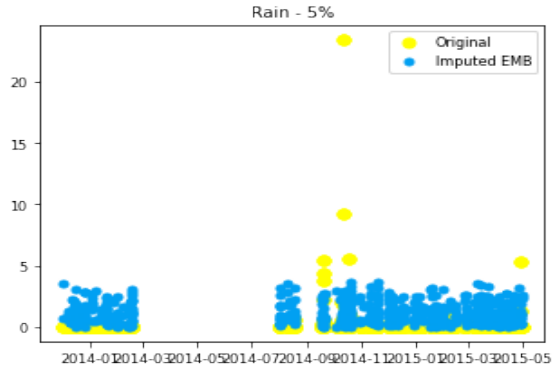
Para una mejor referencia del comportamiento, se puede apreciar la media y las desviaciones estándar en la tabla 6.15



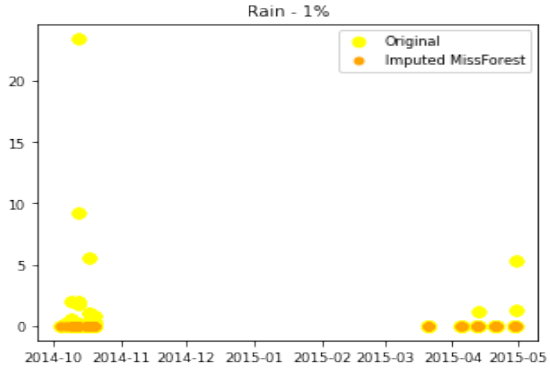
(a) EMB - Rain - 1 %



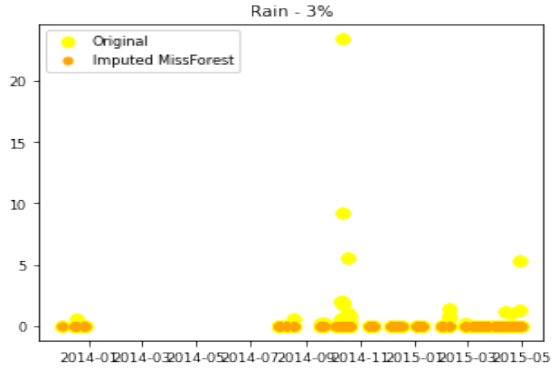
(b) EMB - Rain - 3 %



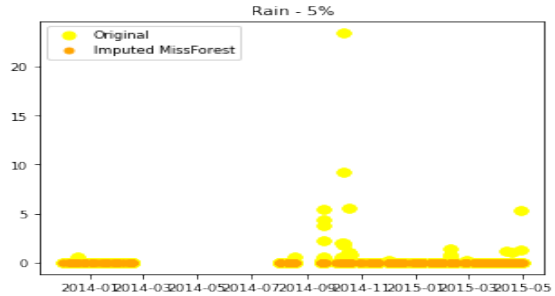
(c) EMB - Rain - 5 %



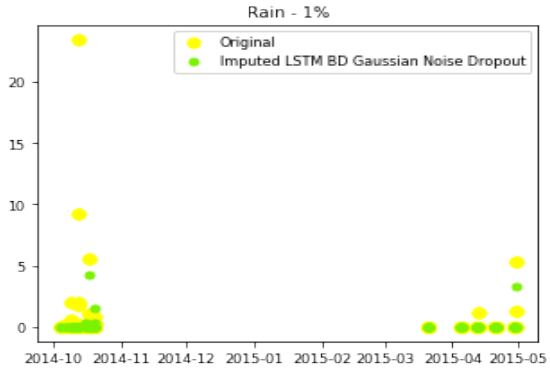
(d) MissForest - Rain - 1 %



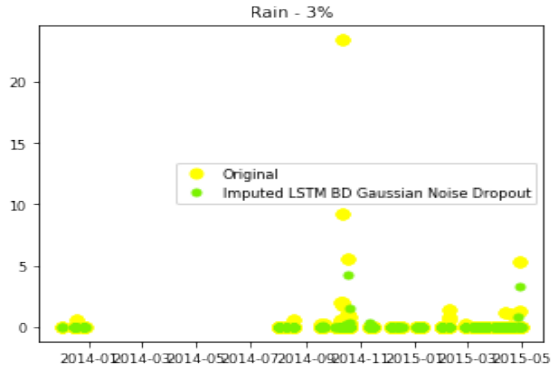
(e) MissForest - Rain - 3 %

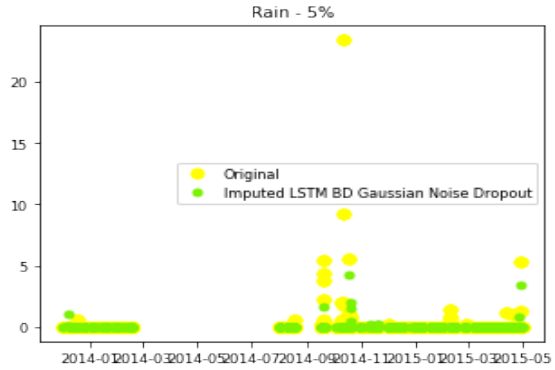


(f) MissForest - Rain - 5 %



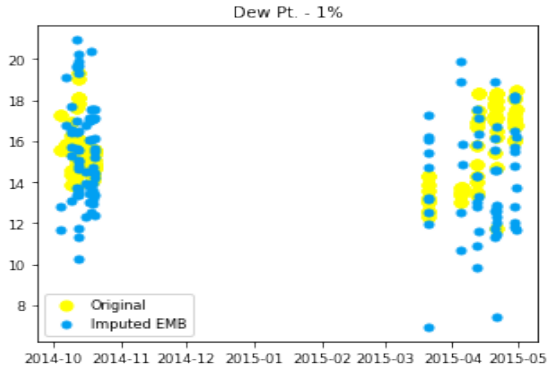
(g) LSTM - Rain - 1 %



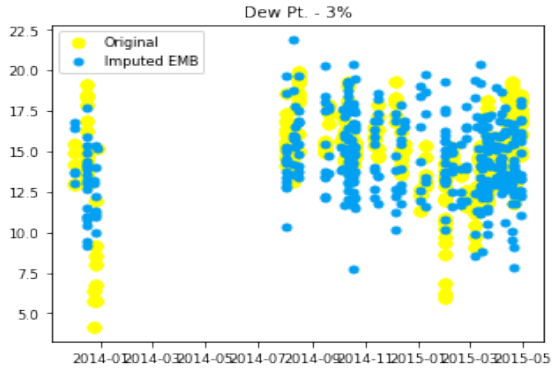


(h) LSTM - Rain - 5 %

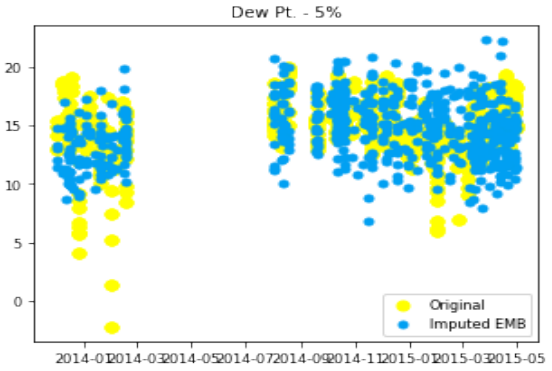
Figura 6.5: Experimento E. Imputación por variable Rain al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la precipitación es en milímetros cúbicos



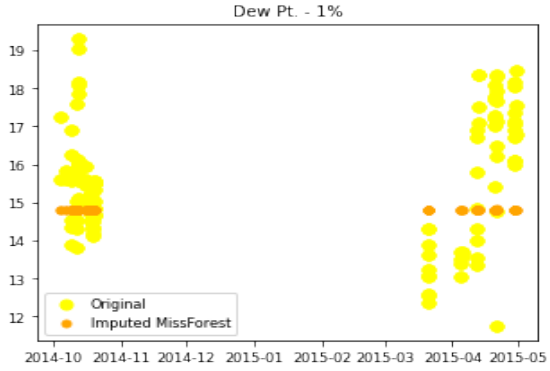
(a) EMB - Dew Pt - 1 %



(b) EMB - Dew Pt. - 3 %

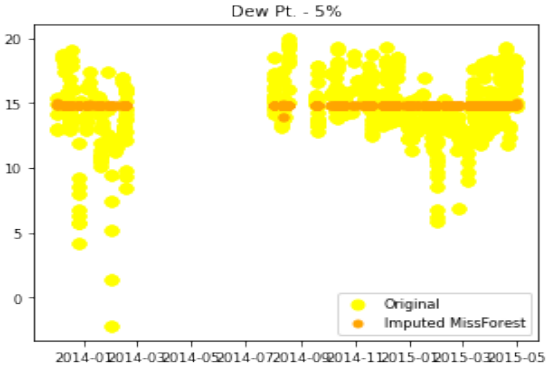


(c) EMB - Dew Pt. - 5 %

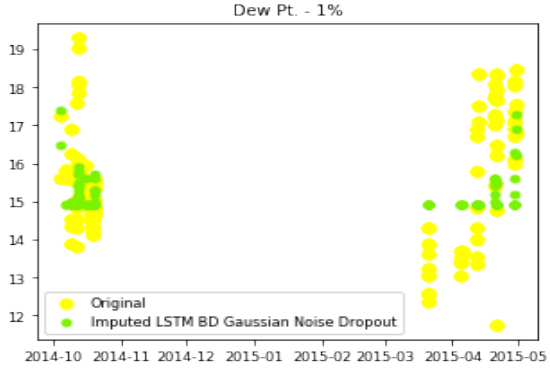


(d) MissForest - Dew Pt. - 1 %

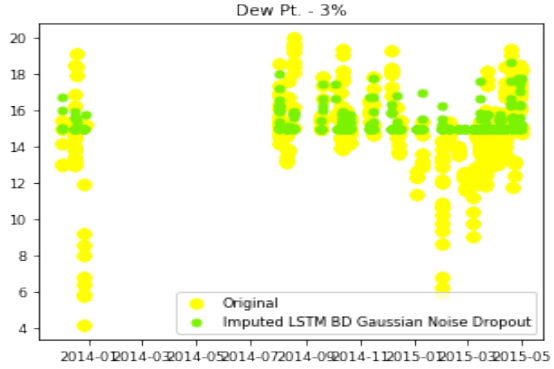
(e) MissForest - Dew Pt. - 3 %



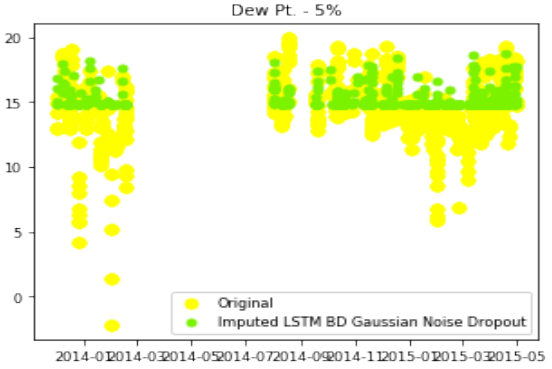
(f) MissForest - Dew Pt. - 5 %



(g) LSTM - Dew Pt. - 1 %

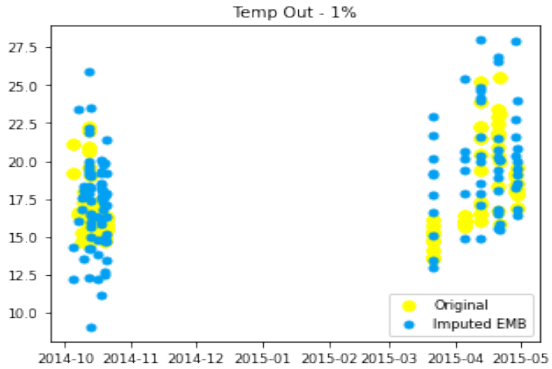


(h) LSTM - Dew Pt. - 3 %

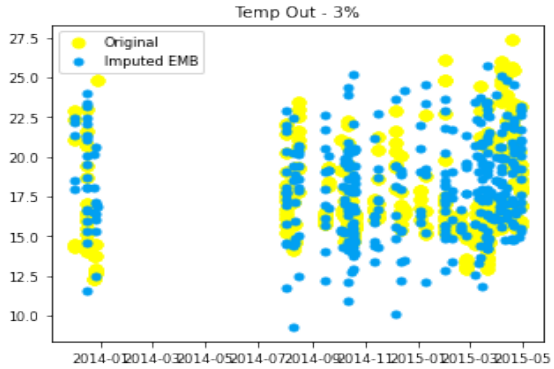


(i) LSTM - Dew Pt. - 5 %

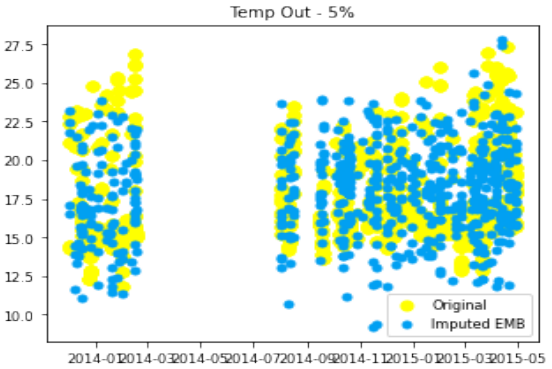
Figura 6.6: Experimento E. Imputación por variable Dew Pt. al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida del punto de rocío es en grados centígrados



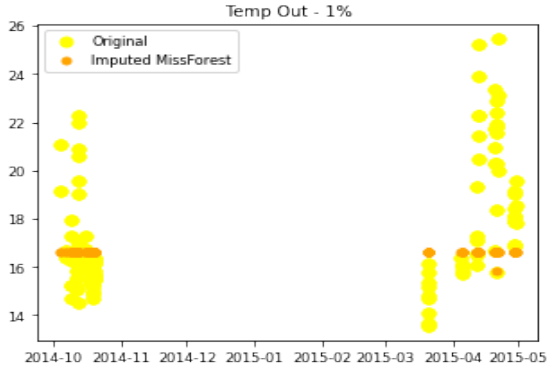
(a) EMB - Temp Out - 1 %



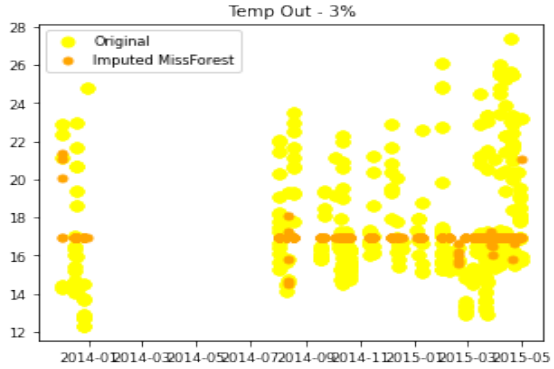
(b) EMB - Temp Out - 3 %



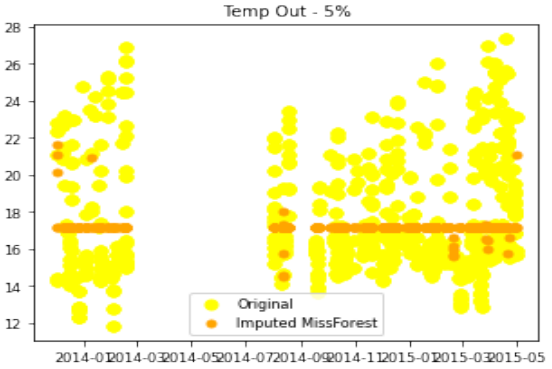
(c) EMB - Temp Out - 5 %



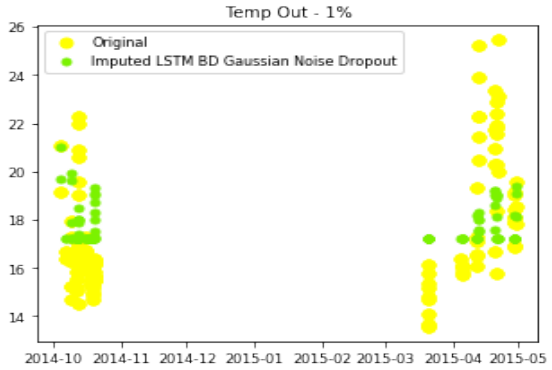
(d) MissForest - Temp Out - 1 %



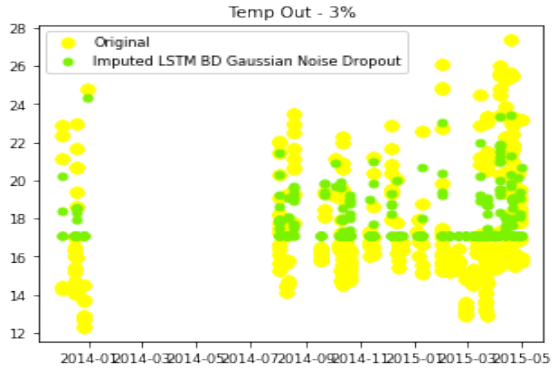
(e) MissForest - Temp Out - 3 %



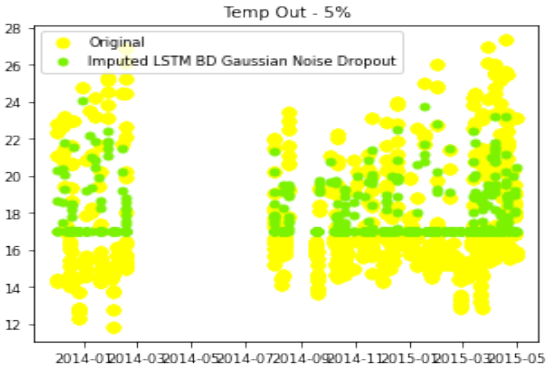
(f) MissForest - Temp Out - 5 %



(g) LSTM - Temp Out - 1 %

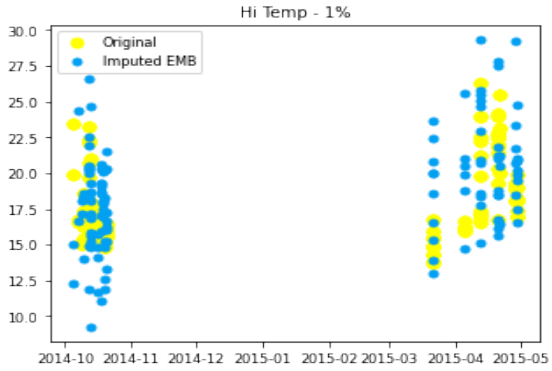


(h) LSTM - Temp Out - 3 %

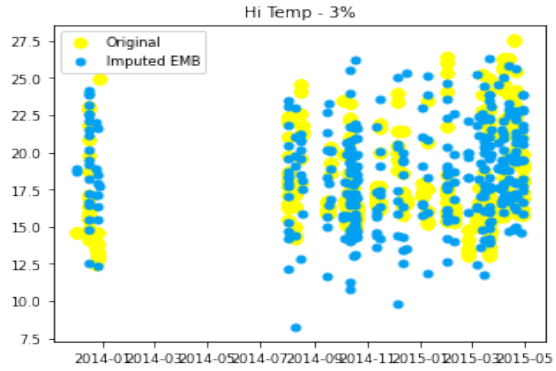


(i) LSTM - Temp Out - 5 %

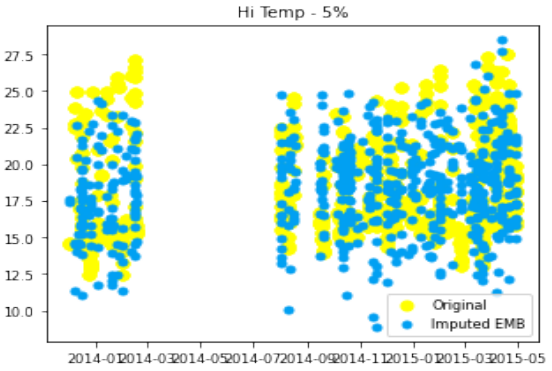
Figura 6.7: Experimento E. Imputación por variable Temp Out al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la la temperatura exterior es en grados centígrados



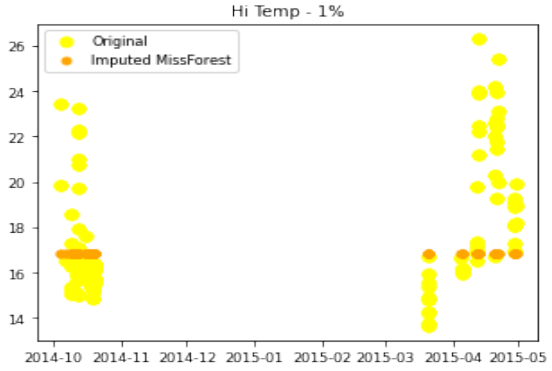
(a) EMB - Hi Temp - 1 %



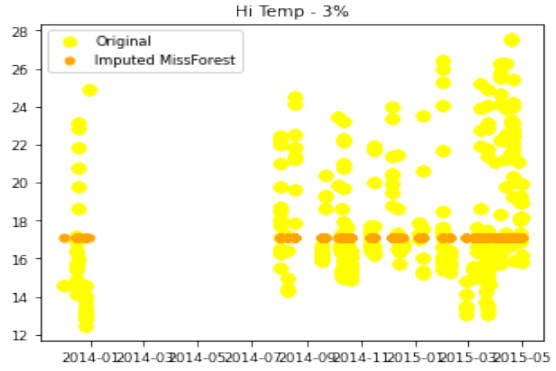
(b) EMB - Hi Temp - 3 %



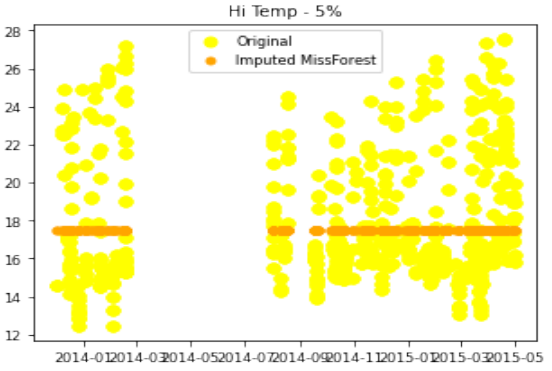
(c) EMB - Hi Temp - 5 %



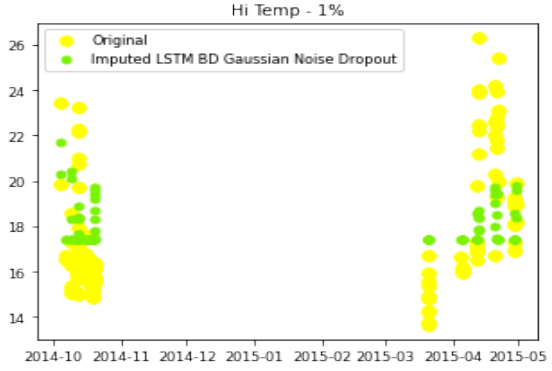
(d) MissForest - Hi Temp - 1 %



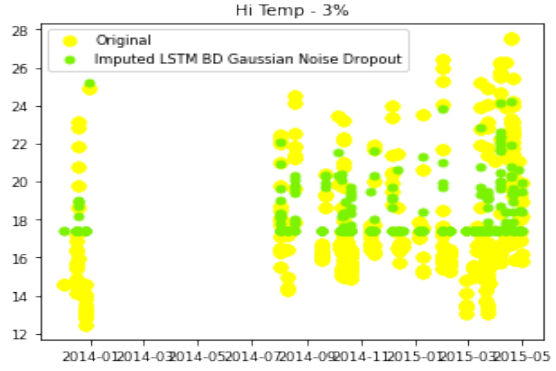
(e) MissForest - Hi Temp - 3 %



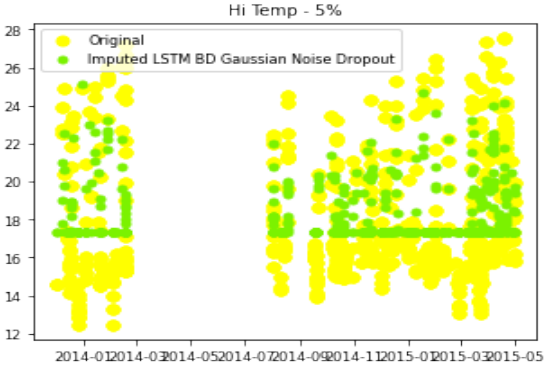
(f) MissForest - Hi Temp - 5 %



(g) LSTM - Hi Temp - 1 %

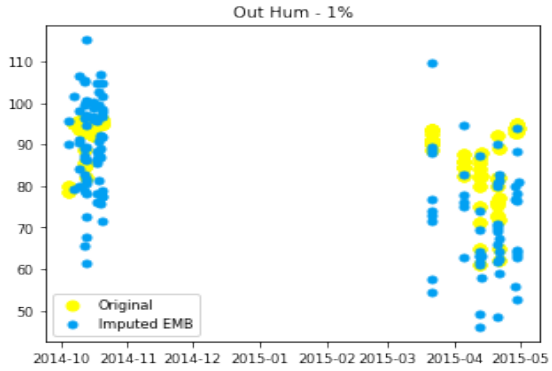


(h) LSTM - Hi Temp - 3 %

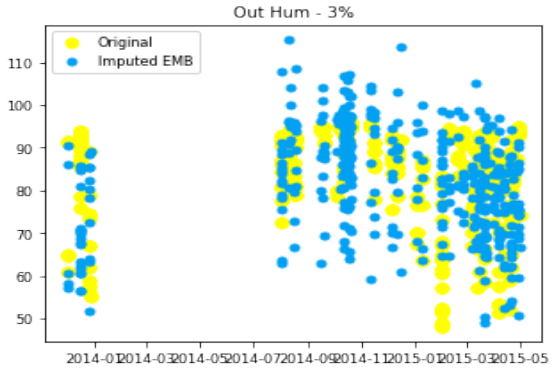


(i) LSTM - Hi Temp - 5 %

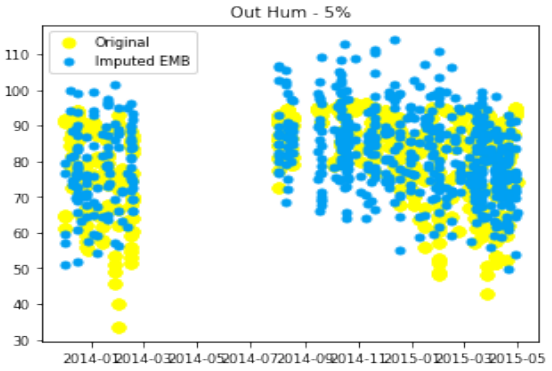
Figura 6.8: Experimento E. Imputación por variable Hi Temp al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la temperatura máxima es en grados centígrados



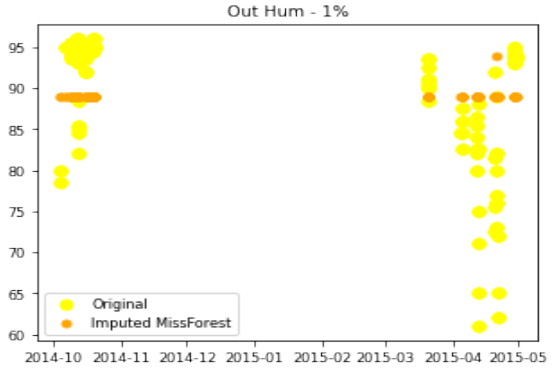
(a) EMB - Out Hum - 1 %



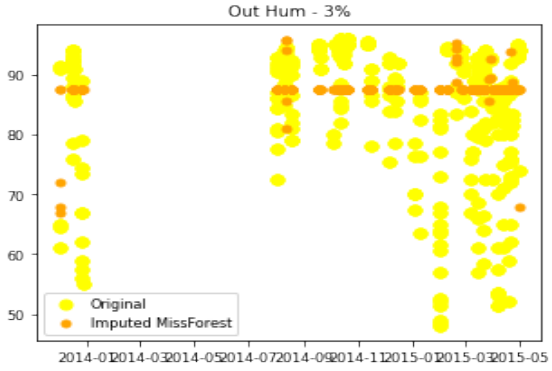
(b) EMB - Out Hum - 3 %



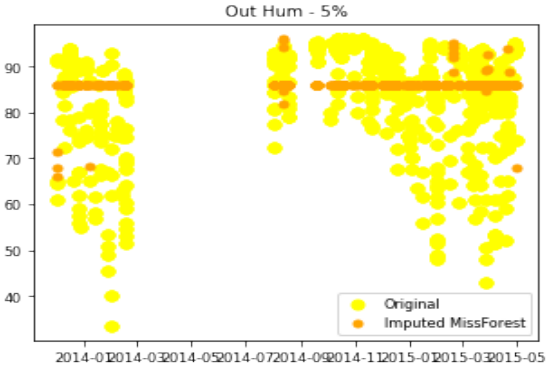
(c) EMB - Out Hum - 5 %



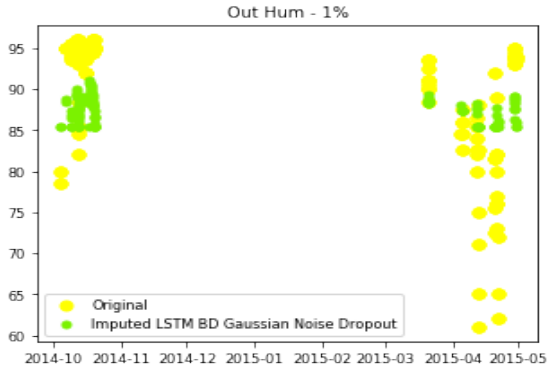
(d) MissForest - Out Hum - 1 %



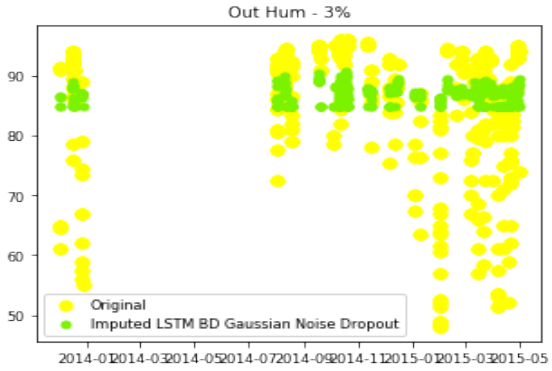
(e) MissForest - Out Hum - 3 %



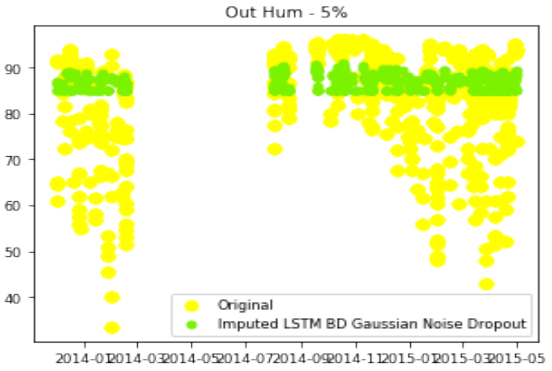
(f) MissForest - Out Hum - 5 %



(g) LSTM - Out Hum - 1 %



(h) LSTM - Out Hum - 3 %



(i) LSTM - Out Hum - 5 %

Figura 6.9: Experimento E. Imputación por variable Out Hum al 1 %,3 % y 5 %. La dimensión X representa el intervalo de tiempo. La dimensión Y unidad de medida de la humedad en porcentaje de humedad relativa

Al visualizar los gráficos presentados por variable en forma independiente, se concluye que:

1. La precipitación 6.5, muestra que EMB se mantiene imputando datos en un solo rango, MissForest no intenta predecir los valores ni acercarse a un aproximado y LSTM para intentar predecir los datos faltantes de una forma más intuitiva el patrón original del comportamiento real de los datos.

2. El punto de rocío 6.6, la temperatura exterior 6.7, la temperatura máxima 6.8 y la humedad exterior 6.9 muestran que EMB es el algoritmo que mejor trata de imputar los datos faltantes, MissForest no intenta predecir los valores ni acercarse a un aproximado y LSTM intenta predecir los datos faltantes de una forma más intuitiva, sin embargo, tiene un límite inferior que no le permite imputar más abajo de sí, además, LSTM busca esas tendencias y estacionalidades para predecir los valores faltantes con mayor exactitud que los demás algoritmos.

3. Se plantea la interrogante de evaluar los mismos algoritmos con la variante de remover la función de activación en la arquitectura LSTM, y así observar su comportamiento, y evaluar el efecto del límite inferior señalado en el punto anterior.

4. También se plantea evaluar cada una de las variables con rangos de aceptación (utilizando una escala de mínimos y máximos proporcionada por ICAFE por variable y a su vez evaluando el sitio web https://www.tiempo3.com/north-america/costa-rica?page=today y determinando cuáles podrían ser los rangos en los que puede variar una variable de una hora a otra hora). De esta manera se establece que

    a) Rain (precipitación): en la mañana cambia poco, como 0.33 mm cada 3 horas, en la noche puede llegar hasta 6 mm. Valor mínimo y máximo aceptable: 0 - 25. Valor de aceptación con respecto al valor original versus el valor imputado: 1 mm.

    b) Temp Out (temperatura exterior): los cambios abruptos cuando entra el

sol y sale, pueden llegar hasta 4 grados. Valor mínimo y máximo aceptable: 10 - 40. Valor de aceptación con respecto al valor original versus el valor imputado: 2 grados.

c) Hi Temp (temperatura máxima): debe ser mayor o igual a Temp Out. Valor mínimo y máximo aceptable: 10 - 40. Valor de aceptación con respecto al valor original versus el valor imputado: 2 grados.

d) Out Hum (humedad exterior): Puede variar en 3 % y cuando entra el sol hasta un 5 %. Valor mínimo y máximo aceptable: 40 - 100. Valor de aceptación con respecto al valor original versus el valor imputado: 3 %.

e) Dew Pt. (punto de rocío): Puede variar en 2 grados y no puede ser mayor que la temperatura. Valor mínimo y máximo aceptable: 10 - 40. Valor de aceptación con respecto al valor original versus el valor imputado: 2 grados.

Para calcular si un valor es aceptado, únicamente se evalúa si el valor imputado cae en el rango de aceptación establecido por variable, en cualquier otro caso dicha imputación es rechazada. Los resultados obtenidos producto de evaluar los algoritmos con rangos de aceptación se pueden visualizar en el siguiente cuadro 6.16.

Es perceptible que dadas los rangos de aceptación descritos, los porcentajes de aceptación alcanzados por LSTM presentan mejores métricas con respecto a las imputaciones realizadas por los otros algoritmos: EMB y MissForest. También la variante del LSTM propuesto sin función de activación en algunas imputaciones de algunas estaciones (específicamente: la estación 9 al 3 % y al 5 %, estación 10 al 1 % y al 5 %, estación 11 al 1 % y estación 12 al 1 %) genera mejores resultados que el LSTM propuesto inicialmente.

EMB	**E9**	**E10**	**E11**	**E12**
1 %	0.40	0.34	0.37	0.30
3 %	0.37	0.33	0.37	0.30
5 %	0.35	0.31	0.35	0.28
MissForest				
1 %	0.48	0.51	0.45	0.43
3 %	0.60	0.50	0.53	0.43
5 %	0.58	0.50	0.51	0.46
LSTM propuesto				
1 %	0.71	0.51	0.66	0.43
3 %	0.66	0.52	0.63	0.44
5 %	0.67	0.54	0.62	0.46
LSTM propuesto sin función activación				
1 %	0.70	0.55	0.71	0.45
3 %	0.68	0.52	0.62	0.44
5 %	0.70	0.55	0.62	0.46

Cuadro 6.16: Diseño Experimental E. Métricas obtenidas al evaluar EMB, MissForest y LSTM con rangos de aceptación.

Conclusiones y Trabajo futuro

En este capítulo se resumen los principales hallazgos, conclusiones y potencial trabajo futuro para esta investigación.

7.1. Conclusiones

La investigación realizada permitió crear una nueva arquitectura de aprendizaje profundo con diferentes evaluaciones de la misma; la primer evaluación basada en una modificación del LSTM que incluye capas bidireccionales con capas de ruido gaussiano y la segunda evaluación con la misma arquitectura pero realizando imputaciones múltiples y multivariadas según el grado de correlación con las estaciones aledañas a ella. La aquitectura propuesta fue evaluada en 4 variantes (utilizando siempre la misma semilla para todos los casos a evaluar):

Los resultados obtenidos en cada tipo de experimento:

1. La arquitectura LSTM propuesta obtiene resultados similares entre las imputaciones realizadas para las estaciones 10, 11 y 12.

2. El algoritmo MissForest obtiene los mejores resultados en los experimentos A (datos faltantes aleatorios), C (duplicando la cantidad de datos para entrenar con respecto al experimento A) y D (generación de nuevo conjunto de datos a partir de las estaciones mejor correlacionadas por cada variable).

3. MissForest no logra imputar datos faltantes en forma consecutiva, siendo el algoritmo EMB la mejor opción cuando los datos faltantes se presentan en intervalos de 2 a 5 días consecutivos en todas las variables (diseño experimental B).

4. El algoritmo LSTM cuando se evalúa en un conjunto de datos de mayor volumen, muestra que, a mayor cantidad de datos para entrenamiento, el algoritmo presenta mejoras en las métricas de imputación. Sin embargo, se cuenta con poca disponibilidad de datos entre estaciones de una misma zona en cantidad de años que nos permitan verificar si con una cantidad de datos de años en forma sustancial, el algoritmo mejora significativamente.

5. Cuando los datos faltantes son aleatorios y a mayor cantidad de datos para entrenar el algoritmo LSTM muestra resultados competitivos con respecto a EMB y MissForest para 1, 3, 5 y 10 por ciento de datos faltantes, sin embargo, para el 20 % de datos faltantes, el LSTM propuesto no brinda resultados competitivos contra los algoritmos en comparación.

6. Cuando se genera un nuevo conjunto de datos a partir de las estaciones mejor correlacionadas con respecto a cada variable los resultados obtenidos no presentaron una mejora importante, debido a que las métricas en LSTM empeoraron a la hora de imputar el 10 % y 20 % de datos faltantes. Al crear un nuevo conjunto de datos donde sus variables no guardan correlación entre sí, se pierde la estacionalidad y tendencias. Sin embargo, sus métricas compiten cuando los datos faltantes son aleatorios (diseño experimental A).

7. Cuando los datos faltantes presentes son aleatorios y continuos a la vez, simulando las fallas reales de los sensores en ICAFE (diseño experimental E), se demuestra que MissForest, a pesar de tener métricas generales aceptables, en realidad no trata de imputar datos, sino que se mantiene imputando en un rango que resulta aceptable, pero que a nivel de visualización por variable, demuestra no estar intentando imputar los datos faltantes en forma intuitiva.

8. Continuando la idea evaluada en el punto anterior EMB muestra utilizar ciertos intervalos y predecir sobre cierto patrón, y es debido a esto es que sus métricas resultan aceptables.

9. Y LSTM (en el experimento E) demuestra estar intentando predecir valores faltantes de forma intuitiva a diferencia de MissForest y EMB.

10. En comparación con los demás algoritmos (experimento E), se realiza un análisis por variable, con rangos y criterios de aceptación o rechazo para cada imputación realizada por los algoritmos LSTM propuesto, LSTM propuesto sin función de activación, EMB y MissForest. Las imputaciones realizadas por LSTM tienen mejores porcentajes de aceptación.

11. Dadas las métricas y la forma de evaluar la hipótesis establecida en la sección del diseño experimental, se rechaza la hipótesis propuesta bajo los criterios evaluados, pues, RMSE no logra mejorar en un 5 % las métricas obtenidas por MissForest y EMB.

7.2. Trabajo futuro

Esta investigación abre las puertas a otras posibles áreas de investigación para mejorar aún más el rendimiento de las arquitecturas de aprendizaje automático con el fin de imputar datos faltantes en series de tiempo o generar nuevos hallazgos en el dominio de la aplicación. A continuación, se mencionan las principales:

1. Encontrar series de tiempo en climatología que contengan una cantidad de datos mayor que demuestren que la cantidad de datos disponibles para entrenamiento en el LSTM propuesto impactan directamente el desempeño del mismo a la hora de imputar datos.

2. Seleccionar variables que guarden un mejor grado de correlación entre sí y que tengan una distribución normal como es lo requerido por el algoritmo EMB para la imputación de datos faltantes o buscar alternativas para variables cuyo

comportamiento no sea normal.

3. En el caso de que alguna estación carezca de alguna información como precipitación o temperatura máxima, nace la interrogante de si la característica faltante podría generarse a través de alguna estación aledaña en su totalidad, a partir de alguna otra característica con la que guarde extrema correlación como se realiza en el experimento D.

4. O bien, una variante a probar podría realizarse generando un conjunto de datos nuevo con las estaciones aledañas mejor correlacionadas por cada variable, pero solamente rellenando los faltantes (y dejando los valores originales) de la estación que se pretende imputar, con esto se genera un nuevo conjunto de datos para entrenar, y a partir de este modelo, se imputa sobre la estación deseada.

5. Generar para los diseños experimentales A, C y D, intervalos de aceptación o rechazo de cada una de las variables y realizar las métricas por variable (al igual que en el diseño experimental E), en lugar de realizar una evaluación general de las imputaciones realizadas.

6. Un ejercicio importante de evaluar para este tipo de diseño experimental A, C y D, sería revisar de la misma manera en que se hizo en el diseño experimental E la visualización de las imputaciones por cada variable para identificar si hay alguna mejora que se pueda hacer a través de la identificación de patrones en las visualizaciones de los gráficos que puedan contribuir en la mejora de la arquitectura propuesta con la arquitectura LSTM.

   Aunque en la sección de conclusiones se rechaza la hipótesis propuesta, se plantea la interrogante de evaluar cada una de las imputaciones por variable y por estación para determinar cuál algoritmo se desempeña mejor y realizar la misma evaluación en los diferentes diseños experimentales presentes en esta investigación con el fin de validar si el algoritmo LSTM propuesto logra mostrar mejores resultados con respecto a los demás algoritmos.

Bibliografía

[1] S. Athey, L. F. Katz, A. B. Krueger, S. Levitt, and J. Poterba, "What does performance in graduate school predict? graduate economics education and student outcomes.," *American Economic Review, Papers and Proceedings.*, vol. 97, no. 7, pp. 512–518, 2007.

[2] A. Chalfin, O. Danieli, A. Hillis, Z. Jelveh, M. Luca, J. Ludwig, and S. Mullainathan, "Productivity and selection of human capital with machine learning.," *American Economic Review, Papers and Proceedings.*, vol. 106, no. 7, pp. 124–127, 2016.

[3] "Illustrated guide to lstm's and gru's: A step by step explanation." `https://towardsdatascience.com/illustrated-guide-skewnesskurtosis2to-lstms-and-gru-s-a-step-by-step-explanation` Accessed: 2022-03-20.

[4] "Time series forecasting with deep stacked unidirectional and bidirectional lstm." `https://towardsdatascience.com/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms` Accessed: 2022-04-28.

[5] "Understanding random forest." `https://towardsdatascience.com/understanding-random-forest-58381e0602d2`. Accessed: 2022-02-08.

[6] J. Honaker, G. King, and M. Blackwell, "Amelia II: A program for missing data," *Journal of Statistical Software*, vol. 45, no. 7, pp. 1–47, 2011.

[7] L. Chen, J. Xua, G. Wangb, and Z. Shena, "Comparison of the multiple imputation approaches for imputing rainfall data series and their applications to watershed models," *Journal of Hydrology.*, vol. 572, no. 7, pp. 449–460, 2019.

[8] N. Bokde, M. W. Beck, F. Martínez-Álvarez, and K. Kulat, "A novel imputation methodology for time series based on pattern sequence forecasting," *Pattern Recognition Letters*, vol. 116, no. 7, pp. 88–96, 2018.

[9] Y. Zhang, P. Thorbun, W. Xiang, and P. Fitch, "A deep learning approach for recovering missing time series sensor data," vol. abs/vH53P-1809, 2018.

[10] Y. Zhang, P. Thorburn, W. Xiang, and P. Fitch, "Ssim -a deep learning approach for recovering missing time series sensor data," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[11] A. Contreras-Juárez, C. Atziry-Zuñiga, J. L. Martínez-Flores, and D. Sánchez-Partida, "Análisis de series de tiempo en el pronóstico de la demanda de almacenamiento de productos perecederos," *ScienceDirect*, vol. 32, pp. 387–396, 2016.

[12] J. J. Miró, V. Caselles, and M. J. Estrela, "Multiple imputation of rainfall missing data in the iberian mediterranean context.," *ScienceDirect*, vol. 197, pp. 313–330, 2017.

[13] A. V. DesherevskiiV, I. ZhuravlevA, N. NikolskyA, and Y. Sidorin, "Problems in analyzing time series with gaps and their solution with the winabd software package," *Izvestiya, Atmospheric and Oceanic Physics*, vol. 53, no. 7, pp. 659–678, 2018.

[14] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: bidirectional recurrent imputation for time series," *CoRR*, vol. abs/1805.10572, 2018.

[15] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer, and J. Stork, "Comparison of different methods for univariate time series imputation in r," 2015.

[16] M. E. Quinteros, S. Luc, C. Blazquez, J. P. Cárdenas-Re, X. Ossa, J.-M. Delgado-Saborit, R. M. Harrison, and P. Ruiz-Rudolph, "Use of data imputation tools to reconstruct incomplete air quality datasets: A case-study in temuco, chile," *Athmospheric Environment*, vol. 200, no. 7, pp. 40–49, 2019.

[17] B. F. Crabtree, S. C. Ray, P. M. Schmidt, P. T. O'Connor, and D. D.Schmidt, "The individual over time: Time series applications in health care research," *Journal of Clinical Epidemiology*, vol. 43, pp. 241–260, 1990.

[18] A. Andiojaya and H. Demirhan, "A bagging algorithm for the imputation of missing values in time series," *Expert Systems With Applications*, vol. 129, pp. 10–26, 2019.

[19] "Australian government." `https://www.statisticshowto.datasciencecentral.com/univariate/`. Accessed: 2019-07-26.

[20] "Investigación descriptiva: análisis de información." `http://cv.uoc.edu/moduls/UW03_84003_01131/web/nwin/m1/medidas_3.pdf`. Accessed: 2019-08-10.

[21] "Univariate, bivariate and multivariate data." `https://www.engineeringintro.com/statistics/introduction-statistics/univariate-bivariate-and-multivariate-data/`. Accessed: 2019-07-28.

[22] "Estadística descriptiva." `http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/EDescrip/tema1.pdf`. Accessed: 2019-08-08.

[23] "Covariables." `https://glosarios.servidor-alicante.com/terminos-estadistica/covariables`. Accessed: 2019-08-11.

[24] A. Eckner, "A framework for the analysis of unevenly spaced time series data," 2014.

[25] L. Campozano, E. Sanchez, A. Aviles, and E. Samaniego, "Evaluation of infilling methods for time series of daily precipitation and temperature: The case of the ecuadorian andes," vol. 5, no. 1, 2014.

[26] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artificial Intelligence in Medicine*, no. 50, pp. 105–115, 2010.

[27] H. James and G. King, "What to do about missing values in time-series cross-section data," *American Journal of Political Science*, vol. 54, no. 2, pp. 561–581, 2010.

[28] Z. Zhang, *Multiple imputation with multivariate imputation by chained equation (MICE) package*, vol. 4. AME Publishing Company.

[29] "Human in machine world." `https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d`. Accessed: 2019-08-08.

[30] "Root-mean-square deviation." `https://en.wikipedia.org/wiki/Root-mean-square_deviation`. Accessed: 2019-08-08.

[31] "Coeficiente de determinación (R cuadrado)." `https://economipedia.com/definiciones/r-cuadrado-coeficiente-determinacion.html`. Accessed: 2019-08-08.

[32] R. P. Díez, A. G. Gómez, and N. D. A. Martínez, *Introducción a la Inteligencia Artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva*. Servicio de Publicaciones, 2001.

[33] "Redes neuronales y aprendizaje profundo." `http://ramanujan25449.blogspot.com/2017/10/redes-neuronales-y-aprendizaje-profundo.html`. Accessed: 2019-08-14.

[34] "Deep learning." `https://la.mathworks.com/discovery/deep-learning.htm`. Accessed: 2019-08-16.

[35] V. F. K. T. e. a. Najafabadi, M.M., "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, 2015.

[36] I. Sucholutsky, A. Narayan, M. Schonlau, and S. Fischmeister, "Deep learning for system trace restoration," *ArXiv*, vol. abs/1904.05411, 2019.

[37] "A guide to rnn: Understanding recurrent neural networks and lstm networks." https://builtin.com/data-science/recurrent-neural-networks-and-lstm. Accessed: 2022-04-28.

[38] "Deep learning for time series forecasting." https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/. Accessed: 2022-04-28.

[39] "How to develop a bidirectional lstm for sequence classification in python with keras." https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/. Accessed: 2022-04-28.

[40] "Rnn and transformers." https://indico.cern.ch/event/982899/contributions/4139452/attachments/2164425/3652700/RNN%20and%20Transformers.pdf. Accessed: 2022-04-29.

[41] "Introducción a redes neuronales recurrentes (rnn)." http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/RNN_LTSM/introduccion_rnn.html. Accessed: 2022-04-29.

[42] "Árboles de decisión con ejemplos en python." https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/. Accessed: 2021-03-21.

[43] "Skymind." https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/. Accessed: 2019-08-14.

[44] C. Zhai., "A note on the expectation-maximization (em) algorithm," *In Course note of CS410*, 2004.

[45] "Tutorial on 5 powerful R packages used for imputing missing values." https://www.analyticsvidhya.com/blog/2016/03/

tutorial-powerful-packages-imputing-missing-values/, march 2016. Accessed: 2019-08-02.

[46] "A gentle introduction to expectation-maximization (em algorithm)." https://machinelearningmastery.com/expectation-maximization-em-algorithm/#:~:text=The%20EM%20algorithm%20is%20an,%2Dstep%20or%20M%2Dstep. Accessed: 2022-02-09.

[47] T. Delleji, M. Zribi, and A. Hamida, "On the em algorithm and bootstrap approach combination for improving satellite image fusion," *International Journal of Signal Processing*, vol. 4, no. 2, pp. 85–86, 2007.

[48] T. Liu, H. Wei, and K. ZhangKey, "Wind power prediction with missing data using gaussian process regression and multiple imputation," *Applied Soft Computing*, vol. 71, pp. 905–916, 2018.

[49] D. H. Schoellhamer, "Singular spectrum analysis for time series with missing data," *Geophysical Research Letters*, vol. 28, no. 16, pp. 3187–3190, 2001.

[50] M. B. Richman, T. B. Trafalis, and I. Adrianto, "Multiple imputation through machine learning algorithms," *Applied Soft Computing*, vol. 71, pp. 905–916, 2018.

[51] D. Montgomery, *Design and Analysis of Experiments*. Student solutions manual, John Wiley & Sons, 2008.

[52] "scikit-learn machine learning in python." https://scikit-learn.org/stable/. Accessed: 2022-04-21.

[53] "P value equal to 0.000 in every test, how is it possible." https://www.researchgate.net/post/P-value-equal-to-0000-in-every-test-how-is-it-possible. Accessed: 2021-09-19.

[54] "Normality-tests-don't-do-what-you-think-they-do." https://www.r-bloggers.com/2011/10/normality-tests-don%e2%80%

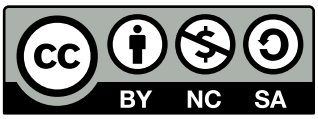
99t-do-what-you-think-they-do/. Accessed: 2021-09-19.

[55] "Testing normality including skewness and kurtosis." https://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/Simon. Accessed: 2021-09-20.

[56] "Assessing univariate and multivariate normality, a guide for non-statisticians." https://core.ac.uk/download/pdf/234680353.pdf. Accessed: 2021-09-21.

[57] Y. Kim, H. Kim, G. Lee, and K.-H. Min, "A modified hybrid gamma and generalized pareto distribution for precipitation data," *Asia-Pacific Journal of Atmospheric Sciences*, vol. 55, 04 2019.

[58] "Lstm and Bidirectional lstm for regression." https://towardsdatascience.com/lstm-and-bidirectional-lstm-for-regression-4fddf910c655. Accessed: 2022-02-10.

[59] "Package 'amelia'." https://cran.r-project.org/web/packages/Amelia/Amelia.pdf. Accessed: 2022-05-31.

Esta obra está bajo una licencia Creative Commons "Reconocimiento-NoCommercial-CompartirIgual 3.0 España".