COLEGIO DE CIENCIAS E INGENIERÍAS

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Planificación para el Desarrollo del Proyecto Integrador

Tutor: Felipe Grijalva

Autor: Alex Pérez

Quito – Ecuador
2024

1. Título del Proyecto

Machine Learning Applied for Cybersecurity of Energy Management Systems
(Aprendizaje Automático Aplicado a la Ciberseguridad del Manejo de Sistemas Energéticos)

2. Relevancia y Justificación

La creciente adopción de fuentes de energía renovable, como los paneles solares, ha traído consigo la necesidad de contar con sistemas avanzados de monitoreo y análisis para garantizar la eficiencia y fiabilidad del suministro eléctrico. Sin embargo, la naturaleza digital de estos sistemas los hace vulnerables a ciberataques, los cuales pueden alterar los datos de generación y consumo de energía, comprometiendo su integridad.

Este proyecto busca abordar este desafío mediante el desarrollo de un modelo basado en redes neuronales recurrentes (RNNs) que permita detectar alteraciones en los datos energéticos reportados por transformadores de paneles solares. La implementación de técnicas de aprendizaje automático permitirá identificar patrones anómalos en los datos y detectar posibles manipulaciones o fraudes. De esta manera, se busca mejorar la seguridad y confiabilidad del sistema energético.

3. Objetivos

a. Generales

Mejorar la precisión en la detección de alteraciones en los datos de electricidad generada por paneles solares, garantizando la integridad y fiabilidad de los sistemas energéticos.

b. Específicos

- Desarrollar modelos con arquitecturas avanzadas para el análisis y procesamiento de mediciones de energía no alteradas.
- Utilizar redes neuronales recurrentes para predecir la cantidad de energía generada y demandada, y detectar posibles manipulaciones.
- Implementar técnicas de aumento de datos para generar muestras adicionales y sintéticas, simulando variaciones en las condiciones de generación de energía.
- Realizar validación cruzada y pruebas de generalización en diferentes conjuntos de datos para asegurar que los modelos sean robustos y funcionen adecuadamente bajo distintas condiciones de generación y consumo de energía.

4. Estado del Arte

Principales Conceptos Relacionados con el Área de Estudio

La actual implementación y adopción de fuentes de energía renovable, como los paneles solares, ha permitido que los sistemas de gestión energética puedan permanecer actualizados con herramientas de tecnología y digitalización. Esto ha permitiendo una mayor eficiencia, sostenibilidad y garantía de que dichas herramientas podrán ser utilizadas en un futuro no tan cercano. Sin embargo, esta digitalización de los sistemas energéticos también los ha hecho más vulnerables a ciberataques. Entre los ataques más comunes destacan los **FDIAs (False Data Injection Attacks)**, que buscan alterar los datos de generación y consumo de energía, comprometiendo la integridad de los sistemas de gestión energética [1, 11]. La identificación y erradicación de estos ataques puede ser posible al utilizar herramientas avanzadas de detección. Dichas herramientas, en la actualidad, pertenecen al campo del aprendizaje automático (Machine Learning, ML), como las redes neuronales.

Con el contexto presentado, el uso de arquitecturas más complejas de redes neuronales ha sido de gran utilidad para la predicción de series temporales y la detección de anomalías en los datos energéticos. Uno de los modelos más utilizados es la **Red Neuronal Recurrente (RNN)** por su eficiencia para trabajar con datos secuenciales. Las RNNs permiten que la información fluya de una neurona a otra a través del tiempo, haciendo posible capturar dependencias temporales [3]. Sin embargo, las RNNs tradicionales sufren de problemas como el "desvanecimiento del gradiente", que las hace ineficaces para capturar dependencias a largo plazo sin el uso de una acercamiento más especializado [4].

Para superar estas limitaciones, se han desarrollado arquitecturas más avanzadas, como las **Long Short-Term Memory (LSTM)**. Las LSTM son una variación de las RNNs diseñadas específicamente para recordar información durante intervalos de tiempo más largos. Las LSTM utilizan "puertas" que regulan el flujo de información, permitiendo a la red decidir qué información mantener y qué información [5]. Esto las hace ideales para la predicción de series temporales, como los datos de generación y demanda de energía, donde es crucial tener en cuenta patrones que pueden ocurrir en distintos periodos de tiempo [2].

Por otro lado, las **Redes Neuronales Convolucionales (CNNs)** han sido ampliamente utilizadas en la clasificación de imágenes, pero su aplicación en series temporales también ha mostrado resultados prometedores [6, 7]. Las CNNs aplican filtros sobre los datos para extraer características relevantes, lo que resulta útil en la detección de patrones espaciales o temporales complejos. Cuando se combinan con LSTM, las CNNs pueden ser eficaces para la detección de FDIAs, ya que permiten extraer características tanto espaciales como temporales de los datos [1].

Recientemente, las **Redes Neuronales Temporales Convolucionales (TCNs)** han emergido como una alternativa sólida a las LSTM. A diferencia de las LSTM, las TCNs pueden procesar series temporales de manera más eficiente, ya que permiten un procesamiento paralelo en lugar de secuencial. Esto las hace más rápidas en tareas que requieren grandes volúmenes de datos secuenciales. Además, las TCNs

utilizan convoluciones dilatadas, lo que les permite capturar dependencias de largo alcance sin los problemas típicos de las RNNs tradicionales [10].

Otra arquitectura que ha ganado relevancia es la de los **Transformers**. Los Transformers se diseñaron inicialmente para tareas de procesamiento de lenguaje natural, pero su capacidad para manejar grandes cantidades de datos y capturar relaciones a largo plazo los ha hecho adecuados también para la predicción de series temporales. Una de las principales ventajas de los Transformers es su capacidad para modelar dependencias de largo alcance sin necesidad de procesamiento secuencial, como ocurre con las LSTM. Esto los convierte en una herramienta poderosa para la predicción de la demanda y generación de energía en sistemas distribuidos [11].

Para gestionar grandes cantidades de datos y entrenar estos modelos avanzados de manera eficiente, se utilizan herramientas como el **DataLoader** y el **DataModule** en PyTorch Lightning. Estas herramientas permiten una carga y procesamiento eficiente de los datos, haciendo más fácil la administración de conjuntos de datos grandes y complejos. Adicionalmente, PyTorch Lightning facilita la estructuración de experimentos, permitiendo un desarrollo más modular y organizado [9].

Por último, la plataforma **Weights and Biases (Wandb)** ha sido utilizada para hacer un seguimiento detallado de los experimentos de aprendizaje automático. Wandb permite a los investigadores visualizar métricas en tiempo real, comparar diferentes modelos y realizar un seguimiento detallado del rendimiento del modelo, facilitando así el proceso de ajuste fino y selección del mejor modelo para la tarea de detección de FDIAs [13].

Investigación de Artículos Científicos Relacionados

Diversos estudios han explorado la detección de FDIAs en redes eléctricas mediante el uso de modelos de aprendizaje automático. Un ejemplo destacado es el trabajo de Zhang et al. (2020), donde se emplean redes neuronales convolucionales (CNNs) en combinación con LSTM para identificar manipulaciones en los datos de generación de energía. Este enfoque híbrido ha demostrado mejorar la precisión de la detección de anomalías, ya que las CNNs permiten extraer características espaciales, mientras que las LSTM capturan las dependencias temporales de los datos [1].

Por otro lado, Wang et al. (2022) investigaron el uso de **Transformers** en la predicción de la demanda energética, lo que facilitó la detección de patrones de manipulación en tiempo real. Los Transformers, con su capacidad para manejar relaciones de largo plazo sin necesidad de procesamiento secuencial, demostraron ser más eficaces que las arquitecturas tradicionales basadas en RNNs para la detección de anomalías en sistemas energéticos distribuidos [11].

Además, Li et al. (2019) aplicaron técnicas de **Redes Neuronales Temporales Convolucionales (TCNs)** para capturar patrones en los datos de energía y detectar anomalías en tiempo real. Las TCNs demostraron ser eficaces al permitir un procesamiento paralelo, lo que mejoró significativamente el tiempo de respuesta

del sistema [10].

5. Metodología de Trabajo

El desarrollo del proyecto se llevará a cabo en varias etapas, comenzando con una revisión exhaustiva del estado del arte, donde se recopilarán los principales conceptos y avances en el uso de técnicas de aprendizaje automático para la detección de ataques FDIA (False Data Injection Attacks) en sistemas energéticos. Esta fase incluirá la identificación de trabajos previos y sus limitaciones, proporcionando una base sólida para el diseño experimental del proyecto.

5.1 Fase Experimental

La fase experimental será implementada utilizando varias herramientas, incluyendo Google Colab, Termius y una VPN para poder acceder a un servidor con mayor capacidad de cómputo. El flujo de trabajo se realiza a través de un port forwarding, que permite conectar Google Colab con un docker ejecutado en un servidor universitario que tiene más capacidad en términos de RAM y almacenamiento. Este servidor se encuentra protegido detrás de la VPN de la universidad, lo que garantiza que solo usuarios autorizados puedan acceder. Una vez conectado, se utiliza un token desde el docker para ejecutar los experimentos en Google Colab de manera remota, aprovechando la capacidad del servidor.

El enfoque principal de esta fase se centrará en la comparación de dos estructuras de predicción:

- **SISO** (Single Input, Single Output): En esta estructura, se entrenan modelos por separado para predecir la demanda y la generación de energía para cada uno de los transformadores del sistema. Se implementarán, principalmente, redes LSTM, TCN y posiblemente Transformers, evaluando sus capacidades de predicción.
- **MIMO** (Multiple Input, Multiple Output): En contraste, la estructura MIMO predice simultáneamente para todos los transformadores, tratando todos los canales de datos de manera conjunta.

Modelos Entrenados

Se desarrollaránn y entrenarán las siguientes configuraciones utilizando Google Colab y la plataforma de experimentación Wandb para el seguimiento y análisis de las métricas de entrenamiento:

- Predicción de la demanda y generación con LSTM.
- Predicción de la demanda y generación con Transformers.
- Predicción de la demanda y generación con TCN.
- Modelo híbrido (LSTM + TCN) para la predicción de la demanda y generación.

Cada uno de estos modelos se entrenará utilizando un conjunto de datos de 25,000 puntos por transformador (el dataset completo tiene un total de 17 transformadores). El primer bloque de datos (0 a 17,500) se utilizará para el entrenamiento, el siguiente bloque (17,501 a 20,000) para la validación, y el último bloque (20,001 a 22,500) para pruebas. Finalmente, los últimos 2,500 datos se emplearán para evaluar el comportamiento del modelo frente a ataques FDIA.

5.2 Detección de Ataques FDIA

El siguiente paso del proyecto es la detección de ataques FDIA en los sistemas energéticos. Una vez entrenados los modelos, se utilizarán tanto las estructuras SISO como MIMO para generar predicciones sobre la demanda y la generación de energía. A partir de estas predicciones, se desarrollará una metodología para detectar anomalías provocadas por manipulaciones en los datos.

La detección de FDIA se llevará a cabo introduciendo perturbaciones manuales en los datos generados. Estas perturbaciones simulan un ataque FDIA, donde los datos manipulados se compararán con los datos predichos. Este proceso permitirá observar diferencias estadísticas entre las características originales y las modificadas.

Para la detección de FDIA, se utilizarán técnicas estadísticas como la distancia media y el test de **Chi-cuadrado**. El sistema manejará un vector de 34 características (17 predicciones de generación y 17 de demanda), lo que permitirá analizar el impacto de las modificaciones en los valores medidos y predichos.

5.3 Análisis de Resultados

El análisis de los resultados incluirá la comparación entre las predicciones generadas por los modelos SISO y MIMO. Se espera que los modelos MIMO, al procesar todos los datos en conjunto, proporcionen una mayor precisión en la detección de ataques FDIA, en comparación con los modelos SISO, que tratan los datos de manera independiente.

Se registrarán métricas de rendimiento para cada arquitectura utilizando Wandb, lo que permitirá visualizar de manera detallada las predicciones y la efectividad de cada técnica en la detección de perturbaciones. Además, se evaluará el tiempo de entrenamiento y la precisión de los modelos en los distintos conjuntos de datos.

6. Sumario de Contenidos

A continuación, se presenta una propuesta inicial de los contenidos que se incluirán en el documento final del proyecto:

- **Introducción**: En esta sección se presentará el contexto y la motivación del proyecto. Se mencionará el objetivo específico y los objetivos generales. Se explicará la problemática de los ataques FDIA (False Data Injection Attacks) en sistemas energéticos y la importancia de utilizar técnicas de aprendizaje automático para su detección.

- **Estado del Arte**: En esta sección se revisarán los principales conceptos, comentarios, observaciones y avances tecnológicos en la detección de FDIA mediante redes neuronales avanzadas, incluyendo LSTM, TCN y Transformers.
- **Descripción de la Propuesta**: Se detallará el enfoque propuesto para la predicción de la demanda y la generación de energía, así como para la detección de anomalías relacionadas con ataques FDIA. Se explicarán las diferentes arquitecturas utilizadas (SISO y MIMO) y la metodología para la generación de datos falsos y la detección vs datos reales.
- **Desarrollo del Prototipo**: Se describirá el desarrollo de los modelos de predicción utilizando Google Colab y Wandb, así como la implementación de las técnicas de LSTM, TCN y Transformers.
- **Experimentos y Análisis de Resultados**: Se presentarán los resultados obtenidos a partir de las pruebas realizadas con los modelos entrenados. Se hará una comparación entre las arquitecturas SISO y MIMO, evaluando su precisión en la predicción y la detección de FDIA. Se incluirán gráficas y análisis de métricas de rendimiento proporcionadas por Wandb y también realizadas de manera propia.
- **Conclusiones y Trabajo Futuro**: En esta sección se presentarán las conclusiones generales del proyecto, destacando las aportaciones realizadas. Se mencionará también posibles mejoras al procedimiento propuesto en caso de que proyectos futuros lo necesiten en diferentes situaciones.

7. Recursos

a. Humanos

- **Estudiante**: Responsable del desarrollo del proyecto, incluyendo la investigación, implementación y análisis de resultados.
- **Tutor**: Felipe Grijalva, quien proporcionará supervisión y guía en todas las etapas del proyecto.
- **Profesores Consultores**: Quienes proporcionaron el dataset utilizado para su estudio y análisis.

b. Materiales

- **Servidor Universitario con Docker**: Utilizado para el procesamiento intensivo de datos y la ejecución de experimentos en paralelo a través de una VPN.
- **Google Colab**: Para la ejecución remota de modelos y la gestión de experimentos a través de la plataforma Wandb.
- **VPN y Termius**: Herramientas utilizadas para el acceso seguro al servidor remoto y la conexión con Docker.

8. Cronograma de Actividades

A continuación se presenta el cronograma tentativo de actividades:

Actividades	Agosto	Septiembre	Octubre	Noviembre	Diciembre
A1: Desarrollo del documento de planificación	x	x			
A2: Estudio del estado del arte	x	x	x		
A3: Implementación de los modelos SISO		x	x		
A4: Implementación de los modelos MIMO			x	x	
A5: Detección de FDIA			x	x	
A5: Análisis de resultados				x	
A6: Redacción del documento final		x	x	x	x

9. Entregables

- **Repositorio en GitHub**: Se proporcionará un repositorio con el código fuente de los modelos desarrollados, la configuración de los experimentos y los scripts para la detección de FDIA. Este repositorio incluirá también la documentación detallada sobre cómo replicar los experimentos, los enlaces a los notebooks de Google Colab utilizados para entrenar los modelos y generar los resultados y el documento en el que se describirán los resultados obtenidos, los análisis realizados, las conclusiones y las recomendaciones para futuros trabajos.

10. Revisión y firma del tutor del proyecto

Yo, Felipe Grijalva, profesor de la carrera de Ingeniería en Ciencias de la Computación, hago constar que he revisado y, por lo tanto, apruebo el documento de planificación del proyecto titulado "Aprendizaje Automático Aplicado a la Ciberseguridad del Manejo de Sistemas Energéticos" propuesto por el estudiante Alex Pérez. Por otra parte, me comprometo a proporcionar al estudiante el soporte necesario y oportuno para el buen desarrollo del proyecto antes mencionado.

Fdo: Felipe Grijalva
Quito, 8 de septiembre de 2024

References

[1] Zhang, Y., Wang, X., & Liu, J. (2020). Detecting False Data Injection Attacks in Smart Grids Using CNNs and LSTMs. *IEEE Transactions on Smart Grid*, 11(4), 3043-3051. https://arxiv.org/pdf/2006.11477.

[2] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673-2681.

[3] Stanford University. (2024). Recurrent Neural Networks cheatsheet. Recuperado de `https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks`.

[4] Cayci, S., & Eryilmaz, A. (2024). Convergence of Gradient Descent for Recurrent Neural Networks: A Nonasymptotic Analysis. *arXiv preprint arXiv:2402.12241*. Recuperado de `https://arxiv.org/abs/2402.12241`.

[5] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv preprint arXiv:1909.09586*. Recuperado de `https://arxiv.org/abs/1909.09586`.

[6] Ankle, L. L., Hegglund, M. F., & Krange, K. (2020). Deep Convolutional Neural Networks: A survey of the foundations, selected improvements, and some current applications. *arXiv preprint arXiv:2011.12960*. Recuperado de `https://arxiv.org/abs/2011.12960`.

[7] O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *arXiv preprint arXiv:1511.08458*. Recuperado de `https://arxiv.org/abs/1511.08458`.

[8] DeepMind. (2022). WaveNet: A generative model for raw audio. Recuperado de `https://deepmind.google/discover/blog/wavenet-a-generative-model-for-raw-audio`.

[9] Stanford University. (2021). Data generation in Keras. Recuperado de `https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly`.

[10] Li, H., Zhang, Z., & Liu, J. (2019). Detection of False Data Injection Attacks in Smart Grids Using Game Theory. *International Journal of Electrical Power & Energy Systems*, 109, 575-581.

[11] Wang, X., et al. (2022). A Survey on Transformer-Based Machine Learning for Energy Demand Forecasting. *Energy and AI*, 3, 100056.

[12] Shervine, A., et al. (2021). Custom Data Generation for Keras Models. Recuperado de `https://medium.com/analytics-vidhya/write-your-own-custom-data-generator-for-tensorflow-keras-1252b64e41c3`.

[13] Biewald, L. (2020). Experiment Tracking with Weights and Biases. *Wandb*. Recuperado de `https://www.wandb.com/`.