Universidad Nacional Autónoma de México

Programa de Maestría y Doctorado en Música

Facultad de Música
Instituto de Ciencias Aplicadas y Tecnología
Instituto de Investigaciones Antropológicas

Resistencias Maquínicas: una caracterización a la improvisación libre con aprendizaje y escucha de máquina

T E S I S

que para optar por el grado de

Doctor en Música (Tecnología musical)

PRESENTA:

Aarón Arturo Escobar Castañeda

Tutor Principal:

Dr. Hugo Solís García, UAM Lerma

Comité tutor:

Dr. Iván Paz, Universitat Politècnica de Catalunya

Dr. Caleb Rascón Estebané, IIMAS, UNAM

México, CDMX. Noviembre 2022

Declaro conocer el Código de Ética de la Universidad Nacional Autónoma de México, plasmado en la Legislación Universitaria. Con base en las definiciones de integridad y honestidad ahí especificadas, aseguro mediante mi firma al calce que el presente trabajo es original y enteramente de mi autoría. Todas las citas de obras elaboradas por otros autores, o sus referencias, aparecen aquí debida y adecuadamente señaladas, así como acreditadas mediante las convenciones editoriales correspondientes.

A Sol.

A Pilar, Arturo y Luanne.

Quiero dar mis mas sinceros agradecimientos por su inagotable impulso e inspiración:

A Hugo Solís, Iván Paz, Caleb Rascón.

Especialmente a Juan Sebastian Lach, José Luis Hurtado, Alfonso Perez.

A Fernando Nava, a Jasmin, a la coordinación del posgrado en música de la UNAM.

A Hernani, Emilio, Tito, Jorge David, Diego, Marianne, Dorian, Xavier, Mike

A Nefi, Diego, Fabián, Gil, Hector, Lucila, Luis, Rossana, Rolando, Milo, Mari Carmen, Amanda, Alexander, Jerónimo, Sarmen, Julia, Julio, Paulina, Emmanuel, Omar.

A Miguel Angel, Toño, Rodrigo, Rafael, Leonardo, Tahuilan, Raul, Daniela.

A las escenas de improvisadores libres de la ciudad de México y el mundo.

A Wade, Chefa, Okkyung, Clare, Eddie, Derek, Tetuzi, Fernando, Remi, Juan Pablo, Wilfrido, Eli, Nicolas, Yan, Mike, Maja, Nick, Otomo.

Índice general

Introducción

"Hay [una] regla que si se transgrede (y más aún si se desvía) es destructora para todo creador: no pongas jamás tu creación al servicio de otra cosa que no sea la libertad. [...] Si la creación tiene un sentido, ese es únicamente el de liberarnos. [...] Piensa en la creación como en una liberación permanente." Estas y otras frases expuestas en el decálogo del cineasta Jan Švankmajerr (Švankmajer *et al.*, 2014) en el cual expone los criterios que sirvieron de guía para sus procesos creativos en su ardua exploración cinematográfica, me llevan a cuestionar cómo y de qué formas busco esa libertad en mi quehacer musical habiendo estudiado una licenciatura en composición. Siguiendo estos cuestionamientos, como creador musical y sonoro, no intento dialogar directamente con aspectos relacionados con la tradición y las formas convencionales de producción musical académica, más bien, busco expandir y sumar herramientas a los modos de mi aproximación creativa, que considero también pueden ser integradas en los recursos de músicos formados o no en la tradición musical académica.

Partiendo de estas búsquedas, la presente investigación surge con el objetivo de entender y expandir las posibilidades creativas de la creación musical, específicamente dentro de la práctica de la improvisación libre, a través de la programación de un *Sistema de Escucha Automática para la Libre Improvisación (SEALI)* desde los procesos complejos generados en la intra-acción humano-máquina. Este sistema toma su conocimiento de múltiples corpus de audio, para "aprender" a escuchar y aplicar ese aprendizaje al análisis tímbrico y estructural de/en la improvisación libre. Derivado de ello, se desprenden algunas características interactivas que le permiten a SEALI generar diálogos coherentes, dotados de algunos elementos sonoros de la libre improvisación e incidir dentro del contexto de ésta u otras prácticas musicales. SEALI surge

de la suma de varias materialidades coexistiendo simultáneamente, siendo la escucha el punto de partida que deriva en las aproximaciones improvisativas humanas, en diálogo con herramientas computacionales enfocadas en el análisis de señales de audio, el aprendizaje automático y lo que supone el aprendizaje e investigación hacia el tema. Dado que el proceso de creación de SEALI fue tomando múltiples caminos durante su desarrollo, me fue posible observar que éste estuvo permeado por un continuo tránsito dentro de un ciclo de retroalimentación que incluye la práctica artística, el desarrollo tecnológico y la investigación. Transitar, explorar y experimentar en este ciclo mediante diversas aproximaciones, hizo evidente que no hay una sola metodología para la resolución de la tarea propuesta, sino que el desarrollo está siempre abierto y en constante afectación debido a la mediación técnica entre humanos y no humanos como señala Bruno Latour en el capítulo *Un colectivo de humanos y no humanos* en su libro *la esperanza de pandora* (Latour, 2001). Sin embargo, dicha afectación que el tránsito por este ciclo propone, tuvo que acotarse para proponer una metodología que se adecuara a los tiempos, objetivos e intereses de esta investigación.

Es así que esta investigación busca abordar desde el concepto de *intra-acción*[1] propuesto por Karen Barad[2], la posibilidad de expandir las dinámicas producidas en la improvisación libre al pensar las intra-acciones humano-máquina no como relaciones independientes sino como agencias con una "exterioridad-dentro" en permanente afectación. Asimismo, busca profundizar en el concepto de "lo libre" desde sus implicaciones dentro de la praxis improvisativa hasta la posibilidad de acceder a su análisis formal/estructural a través de herramientas como las redes neuronales recurrentes (*Long-short Term Memory (LSTM)*) aplicadas al análisis de series de tiempo. Partiendo de ello, la investigación dialoga con seis ejes temáticos fundamentales:

[1] El concepto de intra-acción (Barad, 2007) en vez de interacción alude al enredo (entanglement) de afectaciones que producen los agentes humanos y más que humanos entre sí. La intra-acción problematiza esa dinámica desde el estado de cruce y no solamente desde el resultado del accionar entre.

[2] "La noción de intra-acción (en contraste con la habitual "interacción", que supone la existencia previa de entidades/relaciones independientes) marca un cambio importante, reabriendo y reconfigurando nociones fundamentales de la ontología clásica como causalidad, agencia, espacio, tiempo, materia, discurso, responsabilidad y rendición de cuentas. Una intra-acción específica representa un corte agencial (en contraste con el corte cartesiano, una distinción inherente, entre sujeto y objeto) que efectúa una separación entre "sujeto" y "objeto". Es decir, el corte agencial promulga una resolución "local" dentro del fenómeno de la indeterminación ontológica inherente. Crucialmente, entonces, las acciones internas promulgan la separabilidad agencial, la condición local de la exterioridad dentro de los fenómenos. Así, diferenciar no es una relación de exterioridad radical, sino de separabilidad agencial, de exterioridad-dentro." (Barad, 2011)

- Los entrelazamientos humano-máquina concebidos desde tres dimensiones: el desarrollo algorítmico y conceptual, la investigación y el performance.
- La improvisación libre.
- La escucha profunda (Oliveros, 2005).
- Los criterios de segmentación que toma como punto de partida el concepto del objeto sonoro, entendido como unidades sonoras con cualidades polisémicas (Schaeffer y de Diego, 1996a).
- La escucha y el aprendizaje de máquina aplicados a la detección tímbrica y a la predicción de la forma en la improvisación libre.
- Las técnicas algorítmicas para generar interactividad.

Así pues, el objetivo general de esta investigación es crear un sistema interactivo mediante la escucha y el aprendizaje de máquina, que sea capaz de intra-accionar en tiempo–real en el contexto de la improvisación libre. Esto significa que SEALI sería capaz de reconocer cómo evoluciona una improvisación y desde ese reconocimiento hacer predicciones de lo que ocurre y lo que podría ocurrir subsecuentemente en la evolución tímbrica, espectral y energética de la misma. Esto puede conseguirse a través del análisis profundo de varios grupos de segmentos de audio que responden a distintos niveles temporales: corto, mediano y largo plazo. Esta posibilidad de abordar el aprendizaje sobre distintas temporalidades musicales le da a SEALI la capacidad de aproximarse artificialmente a la idea de gesto, motivo, periodo o sección, partiendo del reconocimiento de estructuras musicales a distintos niveles de profundidad.

Un objetivo particular de la investigación es dotar de herramientas tanto técnicas, como prácticas y metodológicas a creadores musicales, improvisadores, músicos y no músicos así como a desarrolladores de tecnología musical y personas interesadas en abordar y conocer las perspectivas del aprendizaje de máquina vinculado con la escucha artificial y la predicción de series de tiempo. Dada la versatilidad de estas tecnologías, dichas herramientas podrían ser utilizadas, desarrolladas y trasladadas a prácticamente cualquier otro ámbito o proyecto imaginable.

Otro de los objetivos particulares es dar cuenta de los procesos internos, rodeos y derivas llenos de dificultades y aciertos sobre lo que se logró en términos estéticos en todas las etapas del desarrollo de SEALI. Esto con el interés de investigar sobre los aportes que, un sistema interactivo de escucha automática puede proponer dentro de las prácticas de creación musical al momento. De estos desarrollos se desprendieron nueve performances con instrumentaciones y formas de interacción diferentes. Además, son un registro práctico de lo que llamo *espacio de experimentación investigativa* que me permitió abrir un diálogo con improvisadores libres, posibilitando así un espacio de comunicación entre las partes involucradas.[3]

La hipótesis de la investigación parte del siguiente planteamiento: más allá de los cuestionamientos, suposiciones y especulaciones que se pudieran llegar a hacer en torno a si una máquina/sistema-inteligente es creativa por sí misma o no, sugiero que un sistema como SEALI tendría la capacidad de indagar, a través del conocimiento adquirido en su aprendizaje, sobre las formas musicales, tímbricas y espectrales de la música libremente improvisada y podría navegar por diferentes estados de discernimiento. Pensando en que el conocimiento que tendría sobre la improvisación estaría delimitado por una corriente técnica y estética particular, ligada a la práctica de algún improvisador o improvisadores, considero que sería factible hacer predicciones sobre el devenir de una improvisación dada una serie de elementos sonoros iniciales. En este punto resulta interesante pensar en qué tanta relación, manifiesta o no, tendría SEALI con los aspectos característicos del lenguaje musical de uno o varios improvisadores. Esta relación podría entenderse como una transferencia de conocimiento donde todo aquello que coloca al sonido, técnica, o estética particular de una o varias aproximaciones improvisativas, estaría vinculado con lo aprendido y producido por SEALI.

Para comprobar estas suposiciones, planteo las siguientes preguntas de investigación. ¿Cuáles son los elementos a tomarse en consideración para crear un sistema

[3]Cabe señalar que un artículo derivado de esta investigación, así como la de Hernani Villaseñor nos permitió plantearnos el ciclo de retroalimentación que comprende la práctica artística, el desarrollo tecnológico y la investigación. Desde este ciclo, nos damos cuenta de los procesos heurísticos de los que partimos para abordar nuestras respectivas investigaciones doctorales. Este ciclo de retroalimentación fue de suma importancia para ajustar, observar elementos faltantes y calibrar a SEALI a las aproximaciones estéticas, necesidades y técnicas improvisativas de cada uno de los improvisadores implicados en el proceso.

automático que escucha e improvisa con otros improvisadores libres? ¿Qué significa hablar de improvisación libre cuando una máquina, programada para "comprender" esta música, podría replicar ciertos aspectos característicos de la misma? ¿A qué estamos haciendo referencia al hablar de "lo libre" en la libre improvisación? ¿Cómo hablar de forma musical en la improvisación libre, acaso existe el interés de generar forma musical en sus practicantes? ¿Cuáles son los elementos espectromorfológicos de la improvisación libre que dan sentido a la percepción de un performance como este?, ¿en dónde ponen la atención tanto improvisadores como escuchas al momento de crear y presenciar esta música?

Además, dada una secuencia sonora inicial que no necesariamente responde a la base de datos con la que el modelo de aprendizaje de SEALI fue entrenado ¿cómo sucede la transferencia de conocimiento de esta práctica hacia SEALI? ¿Cuáles son las derivas que SEALI propone partiendo de contextos estéticos similares, qué ocurre en el caso de que estos contextos no correspondan, podría negarse a la atención y a la interacción con éstos? ¿Hay algún umbral que le permita activar o no su atención? ¿Cómo son sus sesgos, qué los determina? ¿A cuáles sonidos responde y a cuáles no? Si reaccionara a todo de la misma manera, tal vez agnóstica, siendo reactiva para atender y proponer algo ante todo lo que escucha ¿qué podríamos destacar de esta característica? Solo por intentar emular un comportamiento que le permita dialogar únicamente con un género musical específico, ¿sería necesario activar algún tipo de sesgo para bloquear su actividad y especificidad interactiva a una sola práctica como la libre improvisación? y, en caso contrario, ¿estaría siendo aún más ingenua o perdiendo ámbito para contener en sí otras prácticas musicales? Además, ¿Cómo se relaciona ese sesgo consciente o inconsciente, producto de todas las manos y mentes por las que pasa un algoritmo, con los resultados sonoros que escuchamos? ¿Cómo transforman los algoritmos nuestras prácticas musicales?, ¿cómo participa el artista-programador-investigador en el ciclo que incluye la práctica artística, desarrollo tecnológico y la investigación?[4]

[4]Todas las preguntas aquí expuestas serán problematizadas y respondidas a lo largo de la tesis.

¿Porqué la libre improvisación?

"En el dominio musical, la improvisación no es ni un estilo de música ni un conjunto de técnicas musicales. La estructura, el significado y el contexto en la improvisación musical surgen del análisis, la generación, la manipulación y la transformación específicos del dominio de los símbolos sonoros." (Lewis, 1996) Sí bien como menciona George Lewis, la improvisación[5] no es un estilo, el caso particular de la improvisación libre la considero como un conjunto de prácticas que se caracteriza por emplear elementos y recursos sonoros que tal vez no responden a una aproximación tradicional musical, sino que busca cierta expansión expresiva a través de la creación, modificación e intervención de instrumentos, el empleo de procesadores de efectos, la integración de la música acusmática, el paisaje sonoro o la electroacústica. Además, en ella también se exploran formas de interactuar y cuestionar los procesos creativos imperantes[6] desde la creación individual y colectiva al momento (creación–musical en tiempo–real), el intercambio constante de roles (acompañamiento, solista, integración a un continuo fluir de masas sonoras) y la integración de elementos paramétricos sonoros como son el contenido y la energía espectral, la densidad tímbrica, la complejidad rítmica, la novedad/monotonía entre otros. Además, el empleo de una escucha profunda atenta hacia lo intra y extra-musical es un punto de partida al que muchos improvisadores libres apelan. Es desde los intersticios que se podrían formar de la combinación de todas éstas prácticas que se produce un sistema complejo, en el sentido de (Morin y Pakman, 1998) (Najmanovich, 2008), que da forma a la improvisación.[7]

En los últimos años mi práctica musical se ha centrado en procesos de creación colectiva a través de la libre improvisación, la composición de obras abiertas o modulares y la instalación colectiva. Por otro lado, mi práctica individual como creador

[5] En el capítulo 2 abordaré a profundidad el tema de la improvisación libre.

[6] Por ejemplo, en la composición académica donde el compositor dicta con lujo de detalle la totalidad de los parámetros musicales a considerar, dejando poco o nada de espacio para crear o proponer a el/la/los/las músicos intérpretes. Una perspectiva musical jerárquica donde el compositor se legitima como único agente creativo que requiere que su música sea interpretada.

[7] Al hablar de sistema complejo en esta investigación, hago referencia al paradigma de las ciencias de la complejidad, las cuales parten de la teoría de sistemas para intentar comprender desde una perspectiva holística cómo opera un sistema (social, animal, vegetal, político, artístico, computacional, etc.) a partir de las múltiples interrelaciones que producen diferentes agentes dentro de él. En este caso, la improvisación se concibe como un sistema complejo con sus entradas, variables de estado y salidas que presenta procesos emergentes, homeostáticos y resilientes dentro y fuera de su performace.

musical se sigue perfilando hacia la creación de obras abiertas, las exploraciones electroacústicas procesuales o algorítmicas en vivo, la creación de música para cine desde la práctica de la improvisación dirigida o la "comprovisación", las prácticas vinculadas con la escucha profunda y la grabación de campo. Todo lo anterior, inserto en el marco de la indeterminación, la estocástica, el azar y la emergencia. Cabe mencionar que cada uno de estos procesos abre diferentes posibilidades de creación musical donde se articulan espacios de creatividad muy particulares que generalmente suceden de forma entremezclada y de manera colaborativa. Dentro de esta multiplicidad de prácticas, la improvisación ha representado un centro al que recurro constantemente y que articula a las demás. Por ello me gustaría mencionar a algunos de los improvisadores e improvisadoras que considero representativos dentro de la práctica, que son una gran fuente de inspiración, que me siguen estimulando creativamente y que fueron imprescindibles para el desarrollo de esta investigación, algunos de ellos son: Derek Bailey, Han Bennick, Evan Parker, Pauline Oliveros, Wade Matthews, Chefa Alonso, Okkyung Lee, Clare Cooper, Eddie Prévost, Tetuzi Akiyama, Fernando Vigeras, Remi Alvarez, Juan Pablo Villa, Wilfrido Terrazas, Eli Kesler, Nicolas Collins, Yan Leguay, Mike Majkowski, Maja Ratkje, Magda Mayas, Arthur Henrry Fork, Otomo Yoshihide, Michel Doneda, Paul Rogers, Lê Quan Ninh, Hans Reichel, Alexander Bruck, Fabian Rangel, Milo Taméz, Diego Villaseñor, Nefi Herrada, Alexander Von Schlippenbach, Nicolas Collins, entre muchos otros. Estas prácticas relacionadas al sonido (la improvisación, la composición, la escucha profunda y la grabación de campo) siguen presentes dentro de mi quehacer como músico y creador musical. Lo que busco, parte de la idea de indagar sobre ciertas estéticas fundacionales para intentar proponer una o varias formas de creación musical y sonora que se encuentren en el intersticio entre estas prácticas.

Contribuciones

Con esta investigación busco entretejer una plataforma de comunicación y un espacio de colaboración entre una diversidad muy amplia de perfiles, con roles, objetivos y personalidades tan diversas como las que podrían tener músicos y no músicos, artistas, tecnólogos musicales, investigadores y desarrolladores de código. Parto de la idea de que yo mismo me encuentro en la intersección de cada uno de estos ro-

les en momentos, lugares o situaciones distintas, donde cada uno me ha conducido a desarrollar esta investigación desde una perspectiva trans-nodal. En términos musicales las aportaciones de esta tesis pueden ayudar a compositores, musicólogos e instrumentistas a conocer los ámbitos por los que transita una música en particular, conocer sus perspectivas estéticas a partir del análisis numérico que SEALI provee de las mismas. Además es posible encontrar nuevas posibilidades de reconstrucción musical de un estilo en particular dada una base de datos de algún compositor, o un género musical. En este sentido, las posibilidades que permiten los múltiples desarrollos y metodologías expuestas en esta tesis pueden servir de base para futuras investigaciones o desarrollos de nuevas herramientas para la tecnología musical.

Por otro lado, pensando en la improvisación como una plataforma que permite crear formas muy diversas de comunicación a través del sonido, SEALI puede proveer herramientas tanto a músicos como no músicos para atreverse a improvisar dadas las aportaciones que es capaz de hacer en una performance de improvisación libre. Además considero que no es preciso saber tocar un instrumento para comunicar algo desde una perspectiva sonora. Si partimos de que todos escuchamos, podemos trabajar la escucha profunda como elemento principal de cohesión comunicativa, donde para hablar(sonar), tengo que escuchar al otro. Desde esta perspectiva, SEALI provee tanto una escucha artificial como modos interactivos que pueden ayudar a generar mecanismos comunicativos para la exploración sonora en tiempo–real.

Desde la perspectiva del desarrollo computacional, los aportes de la investigación apuntan hacia el área de la escucha y el aprendizaje de máquina, los cuales pueden utilizarse para analizar y crear modelos computacionales capaces de abstraer a partir de representaciones numéricas, ciertas cualidades musicales o sonoras que pueden aportar al campo de "recuperación de información musical" (Music Information Retrieval) o al análisis de señales digitales, agrupación de datos (clustering) y a la predicción multivariable de series de tiempo en tiempo–real, hallazgos a nivel metodológico con un énfasis directo en el análisis de gestos o frases sonoras significativas independientemente del contexto donde se insertan.

La perspectiva trans-nodal de esta tesis propone desde diferentes frentes, aportar perspectivas que conduzcan a repensar la forma en que nos aproximamos a la creación sonora en general, y a la improvisación libre en particular para cuestionar los automatismos en los que la recursividad de la práctica nos inserta. Además busca fortalecer áreas poco exploradas dentro de la práctica de la improvisación libre a partir de proveer una serie de herramientas de análisis y de interactividad que pueden coadyuvar a la integración de un pensamiento creativo humano-máquina no disociado sino en un continuo fluir que afecta y modifica a cada uno de los agentes implicados en el circuito.

Estructura de la tesis

La tesis está dividida en 5 capítulos: en el primer capítulo "Estado del arte", propongo realizar una caracterización de los diferentes sistemas interactivos lo cual me permite profundizar en el espacio de posibilidades e intra-acciones humano-máquina en la creación musical. Además, abordo algunas de las aproximaciones a la improvisación libre con aprendizaje y escucha de máquina que he encontrado hasta el momento en el que escribo esta tesis y busco el vínculo que tienen estos proyectos con el desarrollo de SEALI. En el segundo capítulo "Escuchar la impermanencia creativa" defino las particularidades de la libre improvisación, expongo las implicaciones que conlleva abordar esta práctica musical, toco ciertos aspectos relacionados con la improvisación libre que me permiten ver algunos de los modos de interacción que surgen dentro de esta práctica y el porqué, en los últimos años, me resulta tan interesante partir de esta forma particular de hacer música en mi quehacer creativo. En el tercer capítulo "SEALI: Sistema de Escucha Automática para la Libre Improvisación" abordo las funcionalidades de este sistema sonoro interactivo que parte de la escucha artificial, modelada desde mi propia escucha, para analizar elementos sonoros de la improvisación libre e interactuar mediante la clasificación y predicción sonora con otros improvisadores. Hablo sobre los múltiples procesos entrelazados por bucles de retroalimentación donde mi escucha y el aprendizaje de máquina se unen con el objetivo de desarrollar la metodología y las funciones computacionales que posibilitan la creación del sistema. En el capítulo cuatro "SEALI: Escuchar la estructura en la música libremente improvisada" presento algunos antecedentes y las

metodologías finales, la implementación de redes neuronales recurrentes, para la predicción de secuencias temporales en el contexto de la improvisación libre. Estas me permitieron abordar las estructuras sintagmáticas de nivel superior como son frases, motivos, períodos y secciones para integrar este conocimiento al sistema e incidir en términos de amplitud, gestualidad y oleadas energéticas en el desarrollo de una improvisación libre. Esta aproximación se extrapola a la capacidad de SEALI de saber en qué momento de la improvisación se encuentra, ya sea el inicio, el intermedio o el final. La propuesta general de este capítulo es abordar la creación de modelos computacionales que contengan la información sobre lo que ocurre en la estructura de la libre improvisación a distintos niveles temporales. Finalmente, en el quinto capítulo "Bitácora de performances con SEALI V.0.1" presento las primeras versiones de SEALI que fueron el punto de partida para proponer un ciclo de siete performances nombrados "resistencias maquínicas" así como la presentación de la versión V.0.2 a través de dos conciertos en el marco del Foro Internacional de Música Nueva Manuel Enríquez. En la primera versión el enfoque se colocó en la identificación en tiempo–real de cualidades y aspectos sonoros que incluyen al timbre, la cantidad de ruido o pureza espectral, la amplitud y el centro espectral del sonido, es decir, los desarrollos expuestos en el capítulo tres. En la segunda se integran los elementos anteriores además de las cualidades para predecir elementos gestuales en la improvisación, es decir, los desarrollados del capítulo cuatro. Asimismo presento algunas reflexiones que me llevaron a pensar el concepto de resistencia como eje nodal para abordar este ciclo y que sirvió como detonador para dar nombre a esta tesis.

Capítulo 1

Estado del arte: Aproximaciones a la improvisación libre con aprendizaje y escucha de máquina

Para contextualizar la temática de este capítulo considero importante plantear la siguiente pregunta: ¿Qué se ha hecho en torno al aprendizaje y escucha de máquina para caracterizar la estructura de la improvisación libre? En esta búsqueda he encontrado que hay pocos trabajos que integren estos tres ejes temáticos además de que un abordaje tan específico como analizar la estructura musical en la libre improvisación, no se ha investigado a profundidad. Por ello a lo largo de este capítulo presentaré algunos sistemas que son similares al que en esta tesis propongo así como algunos otros que emplean herramientas similares a las empleadas en esta investigación. Además de presentar algunos proyectos vinculados con el uso de la inteligencia artificial para la creación musical, buscaré encontrar las relaciones de estos proyectos con el mío. Hablaré de las diferencias entre las aproximaciones al análisis de estructuras musicales partiendo del protocolo MIDI en contraposición con el análisis utilizando señales de audio digital. Abordaré qué elementos son útiles de estos proyectos para

aproximarse al análisis formal de la improvisación libre. Para ello, primero propongo realizar una caracterización de los diferentes sistemas interactivos lo cual me permitirá profundizar en el espacio de posibilidades e intra-acciones humano-máquina en la creación musical.

1.1. Caracterización de sistemas de interacción generativos

Tipos de entradas

Para comenzar con esta caracterización, una primer aproximación sería delimitar lo que puede o no hacer un sistema interactivo para la creación musical. Para ello, la premisa inicial consiste en analizar qué tipo de entradas puede recibir el sistema en cuestión. Si bien aunque podemos hablar de sistemas que podrían tener una capacidad para recibir datos con una resolución que pareciera infinita a nuestra percepción, sabemos que para que un sistema digital pueda interpretar la información de un fenómeno sonoro hay una delimitación escalar de mayor o menor resolución. Ya sea que partamos del protocolo MIDI o, de una descripción numérica del audio (Peeters, 2004), ambos casos usan algún tipo de gradación. Pese a ello, desde nuestra percepción podríamos hablar de una gradación discontinua en el caso del MIDI y continua en el caso de analizar el audio mediante descriptores de audio (Johnston, 1988),(Painter y Spanias, 1997). Aunque sumamente diferentes en profundidad y gradación en la información, ambos paradigmas estarían sujetos a una escala que delimita el tipo de entradas que recibe. Por ello la primer caracterización que destaco son las entradas simples o compuestas.

- Simples: un conjunto delimitado de datos de entrada que no cambian en el tiempo. Un ejemplo sería una entrada MIDI delimitados por tres vectores (amplitud, altura y duración) de 127 posibles valores cada uno. Esta perspectiva podría ser de utilidad para describir numéricamente la música digitalizada en este formato (generalmente música tonal). (Gómez, 2006)

- Compuestas: un conjunto de datos que cambian a una tasa de muestreo fija. Por ejemplo, para analizar la entrada de audio se podrían tomar 16 vectores de

descripciones numéricas de 20 dígitos cada uno. Dependiendo el tipo de descriptores empleados, esta perspectiva puede caracterizar las propiedades de algún objeto sonoro partiendo de su dinámica, tímbrica, altura, espectro, armonicidad, etc. Las relaciones que se pueden producir entre ellas dan lugar a una alta complejidad en la descripción sonora.

Variables de estado/internas

Una segunda característica a considerar serían las variables de estado del sistema, es decir, aquellos aspectos internos que lo componen. Estas características pueden ser muy amplias y dependen enteramente de los objetivos deseados, por ello me limitaré a describir tres aspectos; los modos de procesamiento temporal en los que un sistema puede procesar los datos de entrada, el origen del conocimiento del sistema y sus modos de interactividad.

Modos temporales del procesamiento de los datos

La forma en la que un sistema procesa los datos se puede definir a través de la relación que tiene con el tiempo, puede ser que interactúe en tiempo–real o en tiempo diferido.

- Tiempo–real: cuando el sistema se encuentra disponible de inmediato y es capaz de entregar datos o señales procesadas. Por ejemplo, el procesamiento inmediato de una señal de audio. El tiempo–real también comprende procesamientos de datos en linea, es decir un sistema que no necesariamente responde de inmediato ya que esta procesando datos a través del tiempo, pero que sus funcionalidades están activas sin intervención humana.

- Tiempo diferido: es cuando el sistema esta descontado, o se encuentra inactivo, así se pueden hacer operaciones que no suceden en tiempo–real sino suceden en un tiempo diferente al de la solicitud del procesamiento de los datos. En este caso las activaciones solicitadas suceden en un momento posterior al programado. Esto seria equivalente a la emisión radiofónica o televisiva que no sucede en vivo.

Origen del conocimiento del sistema

- Estático: el conocimiento del sistema esta predefinido a partir de las reglas creadas por el autor.
- Basado en el conocimiento: su conocimiento parte de un conjunto de datos o corpus de información.
- Dinámico/abierto: basado en las entradas que el usuario le proporciona, partiendo de una base previa de conocimiento.

Además, el conocimiento del sistema puede variar en el tiempo o no, ya sea que pueda aprender en tiempo–real o su proceso de aprendizaje sea en tiempo diferido.

La tercer característica sería su modo relacional o forma de interactividad es decir, el tipo de relaciones que establece con agentes humanos o artificiales, a este respecto, detecto cuatro posibilidades:

Modos de interactividad

- Sin interacción con el ambiente externo: Ya sea que procese la señal de audio de forma determinista, sin tener una incidencia que implique acciones en forma de correlación o contrapunto desplegado en el tiempo. Lo más similar a esto sería un procesador de efectos o un *patch* que se activa mediante la intervención humana.
- Interacción con el ambiente externo: cuando el sistema está compuesto por funciones que le permiten actuar de manera autónoma con una persona o grupo de personas. Dada una serie de instrucciones previamente programadas, puede producir contrapuntos o formas de interacción contextual y relacional en el tiempo. Generalmente esto ocurre mediante el intercambio de información y procesos de retroalimentación entre humanos y máquinas.
- Interacción interna: sería un tipo de sistema que interactúa de manera interna con otros componentes de sí mismo siguiendo las reglas correspondientes con

las que opera. Esto puede suceder a través de múltiples agentes artificiales colaborando entre sí para la producción de un resultado sonoro colectivo.

- Interacción híbrida: es un tipo de interacción que sucede cuando varios agentes interactúan proactivamente en el tiempo. Las interacciones se pueden dar entre agentes naturales y artificiales. En esta categorización, sería la mezcla entre la interacción con el ambiente externo y la interacción interna del sistema.

Variables de salida

Finalmente en la cuarta característica están las salidas que el sistema interactivo es capaz de producir partiendo de las entradas que recibe y cómo son procesadas de acuerdo con la programación de sus variables de estado.

- Estáticas: salidas categóricas, generalmente definidas mediante un lenguaje simbólico ya sea una etiqueta numérica o textual.

- Móviles: una serie de variables de salida representadas numéricamente pero que no responden a una asignación categórica sino a una gradación en una escala determinada.

Las salidas que el sistema entrega se pueden asignar, representar o mapear de diversas formas, todo depende de los intereses y objetivos deseados. Si hay un proceso de retroalimentación éstas se tendrían que re-escalar, normalizar o procesar para coincidir con los datos de entrada que el sistema espera. Por lo general para ello se emplea un codificador y un decodificador de la información de entrada y salida.

Además de esta breve y sintética caracterización, para abordar con mayor profundidad una propuesta tipológica de sistemas interactivos recomiendo revisar el apartado 2.3 “Tipología de las máquinas sonoras” de mi tesis de licenciatura. En ésta categorizo una amplia gama de sistemas interactivos, cada uno acompañado de ejemplos artísticos. (Escobar Castañeda, 2016)

A continuación abordaré algunos sistemas interactivos que están vinculados en el cruce del aprendizaje y la escucha de máquina y la creación musical, especificamente, dentro de la práctica de la improvisación libre. Cabe señalar que algunos tocan tan-

gencialmente el tema de la estructura musical y otros la conciben como un resultado emergente producido por las características intrínsecas de los algoritmos.

1.1.1. OMax

OMax es un programa diseñado y desarrollado en el IRCAM por Gérard Assayag y Shlomo Dubnov en 2004. Este sistema aprende en tiempo–real secuencias musicales de longitud variable a partir la escucha automática de improvisaciones libres ejecutadas al momento. De esta escucha extrae información para modelarla y crear una secuencia sintagmática musical (frases musicales), después navega por esa estructura reorganizando los materiales sonoros que son grabados para crear cánones, variaciones y clones que interactúan con el improvisador. OMax integra el algoritmo Factor Oracle propuesto por Allauzen, Crochemore y Raffinot (Allauzen *et al.*, 1999) destacando que es una estructura diseñada para empatar patrones. Este algoritmo es capaz de encontrar estructuras dentro de textos y en el ámbito musical puede generar una correspondencia de patrones musicales. OMax se caracteriza por la identificación acumulada de gestos/frases musicales o sonoras delimitadas, generalmente, por los silencios en la ejecución del improvisador.

El sistema que se ejecuta en Max/MSP puede recibir entradas de audio o entradas MIDI (en una versión de paga y de código abierto respectivamente) y entrega ambas salidas. OMax no utiliza una base de datos precargada, ni tampoco deduce parámetros de la señal de audio. Debido a su programación interna, sus autores lo definen como un sistema agnóstico no sabe nada al respecto de estéticas musicales y por ello puede adaptarse fácilmente a diferentes contextos. (Levy, 2012)

Experiencia al improvisar con OMax

OMax en su versión MIDI presenta cuatro módulos generales: la entrada MIDI, el procesamiento de los datos, la salida y la visualización de los motivos ordenados por frases que van siendo grabados y después reproducidos. Las pruebas que realicé consistieron en enviar datos MIDI desde un piano digital a OMax, mientras OMax enviaba de regreso su salida al mismo piano digital. Al iniciar la improvisación me encontré con un *clon* de los gestos que propuse, ya que la interacción del sistema parte

de la noción de repetición, por momentos literal, en otros, aplicando ligeras variaciones a lo tocado. Al seguir improvisando me encontré con que OMax reproduce de forma seudo aleatoria algunos fragmentos previamente tocados, generando con ellos, una suerte de memoria del sistema. Con ello, al sistema no le importa en qué estilo vaya la improvisación, que tonalidad tenga o cuál sea su densidad sonora, simplemente graba e inmediatamente reproduce, por lo que se amalgama rápidamente a la improvisación. Esto puede tener sus ventajas si pensamos a Omax como una suerte de extensión del instrumento o del mismo improvisador.

Desde mi experiencia, en un principio encontré un poco limitado interactuar con OMax debido a su inequívoca memoria de máquina que imita los gestos propios integrando ligeras variaciones (estas incluyen modificaciones rítmicas, en las alturas y en la velocidad de reproducción). Siendo improvisador libre enfocado principalmente en el ruidismo, fue todo un reto probar esta versión del programa. Primero por los límites diatónicos que impone la retícula del teclado, del cual se extraen la altura y velocidad dinámica que después son usadas por el sistema. Segundo, en la configuración que usé para tocar pareciera que uno mismo estuviera tocando esos gestos nuevamente. De manera tal, que hay la posibilidad de acceder a una especie de memoria personal digital que recuerda cada gesto tocado. Además, la similitud con que reproduce los fragmentos en el orden en que fueron tocados y la linealidad de la reproducción es tal, que, por momentos crea una suerte de canon imitativo monótono. Algo interesante fue que al introducir un nuevo material, el sistema regresaba los primeros gestos con los que yo había comenzado a improvisar. Esto empuja la improvisación a una interacción y forma estructural que puede ser predecible después de un tiempo y por momentos algo limitada debido a la repetición literal de los mismos gestos.[1] En este sentido, desde mi experiencia no hay una propuesta innovadora que rompa con lo que el improvisador proponga, sino más bien un tipo de "empalagamiento" sonoro.

Sin embargo, el patch de Max/msp tiene la posibilidad de elegir entre tres modos distintos de improvisar, cada uno con distintas propiedades que pueden hacer un poco más rica la interacción si estos y sus diferentes parámetros son manipulados por otra persona. Las posibilidades de asignar distintas velocidades de reproducción a cada

[1] `https://www.youtube.com/watch?v=zm5td1aSsug` Fecha de consulta 29 de noviembre 2021.

uno de los modos de improvisación, de activarlos y desactivarlos, alterar las transposiciones al momento, cambiar aleatoriamente el lugar de reproducción o asignar las salidas a distintos timbres fueron elementos que indudablemente enriquecieron el panorama reactivo en la interacción del sistema.

En un principio los resultados pueden dar la impresión de un comportamiento "orgánico", dado que el programa detecta correctamente los gestos musicales y se mantiene dentro del mismo estilo propuesto por el improvisador, en este sentido, no hay una respuesta que esté completamente fuera de lugar. Sin embargo, en algún momento deja de hacer falta que el improvisardor siga tocando ya que OMax tiende a acumular gestos de forma lineal sin que el improvisador pueda controlar la acumulación y densidad de los materiales. De modo que después de unos seis minutos deja de existir una sensación de diálogo, tornando el proceso en un monólogo de la máquina, que toma el control y se convierte en el protagonista de la improvisación. Por otra parte, no hay nada que el mismo sistema proponga por sí mismo respecto a lo cual uno pueda reaccionar. Después de un tiempo, OMax no puede regresar a las primeras partes de la improvisación para tocar materiales previos e intentar proponer nuevas tendencias en la improvisación, por lo que podría entenderse como un sistema lineal y cerrado. Lineal por su predictibilidad interactiva y cerrado porque no puede proveer o aportar conocimiento nuevo en términos musicales. OMax no establece en ningún momento un rol de interacción ya que solo lanza gestos automáticamente, dejando de lado el intercambio que podría producirse en una improvisación mediante una escucha activa de la cual podrían emerger nuevas propuestas sonoras. El sistema finalmente impone su forma de operar, por lo cual, la manera en la que uno podría interactuar está completamente mediada por sus rasgos interactivos y formas de reacción limitadas. Lo verdaderamente interesante en un sistema de improvisación de este tipo sería su capacidad de proposición y adaptación a distintas situaciones, procurando tomar un papel proactivo a través de la escucha activa de varios parámetros musicales y no solo reactivo que supone una escucha pasiva, mecánica, determinista; que nos lleva a pensar en que la máquina está interactuando cuando realmente se trata de una ilusión. Al emplear sistemas interactivos proactivos se podría generarse

una provocación que active ideas o exploraciones más allá de lo que un improvisador conozca o pueda proponer.

A este respecto, Schankler, Smith, François y Chew en su artículo sobre *factor Oracle*, mencionan que el algoritmo de reconocimiento de frases musicales en OMax podría presentar características intrínsecas a su programación que predisponen al algoritmo a obtener resultados estructurales particulares en su análisis. Independientemente de lo que toque el improvisador, el sistema termina por imponer sus propias formas de interpretar la información de entrada y de manera tal, sus formas de improvisación que la mayoría de las veces serán muy similares.(Schankler *et al.*, 2011)

Por otra parte, en performances que he encontrado con OMax en su versión de procesamiento de audio (en vez de la versión MIDI), encuentro varias ventajas al improvisar. El sistema muestra una capacidad de adaptación mayor al contexto sonoro de la improvisación. Esto debido precisamente a la capacidad de grabación y reproducción del sistema y la interacción que se genera. Ejemplo de ello lo podemos encontrar en la performance de la nota al pie, entre un improvisador (fagot), un performer/controlador (humano) de OMax y el propio sistema OMax.[2]

Pese a las ligeras transformaciones que OMax realiza a los gestos sonoros grabados y las intervenciones casuales que tiene sobre el proceso de improvisación del fagotista, de las cuales emergen ciertas tendencias estilísticas afines, OMax no tiene forma alguna de adaptarse a lo que está ocurriendo. Es a través de factores emergentes, el azar y de las decisiones del performer que controla a OMax, que la interacción global deja escuchar cierta evolución, en términos de densidad y dinámica, aludiendo a la estructura de la campana de Gauss. Es decir, comenzar el discurso con pocos materiales sonoros, desarrollarlos y llevarlos a un clímax en cuanto a densidad, brillo o volumen y finalizar *al niente*. En este sentido, al respecto de las estructuras formales los autores postulan que:

> estos patrones estructurales, si se observan consistentemente, podrían ser propiedades emergentes del comportamiento del factor Oracle [...], algunas de las estructuras formales observadas en las improvisaciones grabadas

[2] `https://www.youtube.com/watch?v=pojhhJN1ySE`. Fecha de consulta 29 de noviembre de 2021.

> [...] surgen, al menos en parte, del comportamiento [...] inherente [del algoritmo]. (Schankler *et al.*, 2011)

Pese a sus limitantes, OMax privilegia ciertas características interactivas sobre otras; por un lado la apertura técnica que podría tener hacia la libre improvisación, pero por otro, se olvida completamente de la estructura global de la misma y de la capacidad de escuchar activamente al otro para establecer un diálogo que le permita adoptar otros roles de interacción dentro de la improvisación.

1.1.2. GREIS

¿Qué ventajas tendría improvisar con un sistema que distingue tendencias sonoras durante el performance frente a un sistema previamente entrenado sobre un corpus musical y/o sonoro? Partiendo de esta pregunta introduzco las ventajas que encontraron los creadores de GREIS (Sistema de Instrumentos Expandidos con Retroalimentación Granular), el cual aprende en vivo a través de una escucha humana "prestada" durante una improvisación libre. Al respecto los autores comentan:

> Si bien consideramos que el pre–entrenamiento es una dirección importante e interesante a tomar [en la creación] de sistemas interactivos [...], hasta la fecha nos hemos centrado en sistemas que aprenden tendencias sonoras y estilísticas durante el performance para privilegiar la naturaleza abierta de la improvisación completamente libre, así como para ver hasta qué punto se puede llegar con este enfoque.(Van Nort *et al.*, 2013)

GREIS fue generado de forma colaborativa por Pauline Oliveros, Jonas Braasch y Dough Van Nort en 2013, cada uno aportó lo más significativo de su campo para crear un sistema que improvisara y pudiera adaptarse a través de una escucha profunda a diferentes contextos de la libre improvisación. El sistema no usa la tecnología de máquinas que aprenden, en cambio es una simbiosis entre sistema y performer, quien se convierte en "los oídos" de la máquina, éste tiene que detectar y grabar en vivo momentos que considera interesantes de la improvisación para que después sean almacenados en la memoria del sistema. Por ello, la máquina no escucha, ni reconoce contextos, sino que requiere de un intérprete que escuche por ella y pueda anticipar

ciertas acciones sonoras/musicales de los otros improvisadores y del mismo sistema para ayudar a conducir el performance. GREIS puede reaccionar y tomar decisiones en tiempo–real partiendo del análisis y mapeo espacial de las muestras de audio almacenadas que posteriormente serán moduladas por la máquina y regresadas en forma sonora a los improvisadores.

Esta memoria capturada se puede mantener de forma indefinida o hasta que sea borrada de forma intencional, permitiendo moldear el contenido de audio almacenado también de forma manual por el intérprete.

> El intérprete tiene un control refinado de las inflexiones gestuales pasadas, conduciendo [la improvisación] hacia un diálogo con intenciones gestuales pasadas y futuras, así como con variantes introducidas por la máquina [...] Esta propagación de la intención gestual es clave para que el sistema sea capaz de definir una estructura musical cohesiva que fluya orgánicamente [...]. (Van Nort *et al.*, 2013)

El sistema tiene una memoria episódica según sus autores, esta es capaz de registrar, además de audio, parámetros de control que después pueden usarse para representar y modular el contenido sonoro producido por el sistema, cambiando con esto las intenciones musicales y las modulaciones del intérprete hacia el sistema y viceversa. Además GREIS agrega al performance elementos indeterminados producto del ruido en la transmisión de la información generado por la captación, análisis, transformación y reproducción sonora, agregando a las improvisaciones, variantes completamente insospechadas. Además,

> Con el fin de mantener el carácter sorpresivo en la improvisación, el sistema de instrumento expandido [EIS, GREIS], cuenta con una capa de control de alto nivel de aleatoriedad dirigida, introduciendo ruido en los parámetros de procesamiento sonoro, así como la matriz de enrutamiento y filtrado por grano, mientras que el comportamiento de las líneas de re-

> tardo moduladas pueden controlarse de forma probabilística. (Van Nort *et al.*, 2013) [3]

GREIS como instrumento expandido tiene muchas posibilidades y ventajas que pueden ser usadas en el performance, sin embargo como se ha mencionado, no cuenta con un sistema que le permita escuchar y adaptar su escucha de forma inteligente a los distintos escenarios de la improvisación. Justamente estas limitantes son las que posibilitaron crear a su sucesor FILTER, "el cual puede concebirse como una expansión del instrumento expandido en el territorio de la escucha de máquinas y una aproximación evolutiva hacia la creación de salidas musicales"(Van Nort *et al.*, 2013)

1.1.3. FILTER

Para sus autores, FILTER se puede amalgamar de manera sorpresiva con las intenciones de los músicos debido al grado de interconectividad generada. De manera tal, que por momentos resulta difícil saber quién es el que realmente está tocando, si la máquina o los músicos. Las casi infinitas posibilidades de combinatoria tímbricas entre un ensamble, además de las exploraciones técnicas usadas, vuelven sumamente difícil poder reconocer con claridad los materiales sonoros. Al escuchar algunas improvisaciones realizadas con FILTER, por momentos es difícil saber con certeza quién está produciendo un determinado sonido en la improvisación.[4] Pauline Oliveros, Jonas Braasch y Dough Van Nort comentan:

> Como resultado del intercambio de señales sonoras, a veces, el resultado son tres gestos sonoros distintos con cualidades tímbricas únicas; otras veces estas formas gestuales pueden tener timbres muy similares. En otros momentos el trío se fusiona en una sola línea moviéndose de forma coherente y da como resultado una textura sonora que se sostiene sin dirección lineal. Las decisiones musicales improvisadas en el performance se guían puramente por la escucha focal y global en el momento, y ambas son

[3] Al hablar de la matriz de enrutamiento los autores se refieren al espacio en donde están almacenadas las muestras de sonido, así mismo el audio es dispuesto para ser procesado mediante síntesis granular.

[4] `https://web.archive.org/web/20160829121542/http://dvntsea.weebly.com/sound.html` `https://soundcloud.com/dvntsea` Fecha de consulta 29 de Noviembre 2021.

> conscientes del movimiento entre la dinámica gestual y el sostenimiento textural.(Van Nort *et al.*, 2013)

Al igual que GREIS, FILTER graba materiales tocados por los improvisadores, lo cual resulta en una amalgama de sonidos inevitable e inherente a su programación. Determinar a partir de la escucha las acciones concretas de cada músico incluyendo las de los dos sistemas,[5] no sería una tarea fácil para una escucha normal, por más profunda que esta sea. Por lo cual tuvieron que hacer varias modificaciones al sistema ya que en algunos momentos los autores buscaban tener claras las intenciones y acciones del humano, y diferenciar las de la máquina.

FILTER también registra la evolución temporal de los componentes sonoros analizados, con esto genera el nivel semántico y estructural de su memoria la cual contiene un conjunto de materiales sonoros dispuestos en frases, generalmente divididos por silencios o respiraciones. Estas características de aprendizaje e incluso la forma de interactuar con otros músicos están mediadas por la forma en que fue "enseñada" a escuchar.

Del análisis correspondiente de los componentes extraídos de cada gesto, FILTER aprende con un modelo de aprendizaje sin supervisión, el cual agrupa de forma automática los diferentes sonidos introducidos en su sistema y los organiza por clases. Cuando un cambio de amplitud o frecuencia es detectado, sujeto a una compuerta temporal de ataques (onset threshold), se considera que un nuevo gesto o frase ha comenzado. Si el gesto detectado es diferente a los que están dentro del banco de gestos, será incluido como un nuevo miembro del banco sonoro. Este tipo de comportamiento es lo que los autores consideran como el nivel semántico de la memoria del sistema, capaz de representar objetos abstractos de la improvisación, con este nivel compara entre su banco de sonidos y todos los sonidos entrantes. Su función específica es encontrar dichos gestos sonoros relevantes al momento de la improvisación. La memoria semántica de FILTER no tiene ningún tipo de ordenamiento temporal entre los gestos, es decir que estos se clasifican indistintamente del momento en el

[5]La formación del trío con la cual probaban, ensayaban y se presentaban en vivo incluía a Pauline Oliveros acordeón, Jonas Braasch saxofón, Dough Van Nort manipulando GREIS y FILTER funcionando de forma completamente autónoma.

que suceden, a este respecto podría considerarse que la memoria semántica se encuentra fuera de tiempo. Esta podría ser una ventaja del algoritmo, que para fines de interacción en la improvisación libre podría resultar adecuado, si lo que se busca es conservar solamente los tipos de materiales similares para ser tocados en la evolución de la improvisación.

Además en una escala temporal más amplia, el sistema realiza cortes entre segmentos que tienen diferencias sustanciales texturales y tímbricas, de ellos extrae un promedio de los parámetros tocados al mismo tiempo por todos los intérpretes, este promedio será traducido como la textura general del ensamble y es entendida como la memoria estructural del sistema. Esta memoria será utilizada en momentos en donde el algoritmo de reconocimiento no alcance el puntaje requerido para detectar de forma certera gestos musicales individuales y accionar sus modos de reacción. Sus reacciones son el resultado de la calidad del análisis, de esa escucha profunda artificial, entrenada y programada por los autores.

Con las cualidades anteriores como se ha mencionado, FILTER compara los segmentos de entrada con los segmentos gestuales contenidos en su banco de datos, "proveyendo un vector de posibilidades para cada miembro [sonoro almacenado]. Mientras un sistema de clasificación estándar miraría hacia el miembro más parecido, nosotros examinamos la probabilidad completa de los vectores y cómo cambia su forma dinámica en el tiempo."(Van Nort *et al.*, 2013) De este modo podría decirse que FILTER cuenta con un sistema de atención dinámica, la cual identifica al momento de la improvisación ciertos arquetipos gestuales y texturales comparándolos con los almacenados en su memoria.

La metodología usada en la fase de escucha fue grabar gestos en dos escalas temporales una amplia que distingue la textura y estructura general de la improvisación y una escala temporal corta que distingue entre gestos musicales/sonoros, después se entrenó al sistema para reconocerlos al ser tocados nuevamente. El sistema especifica el grado de similaridad y certeza con que fue reconocido un gesto, posteriormente ciertos rangos de certeza en el reconocimiento de las frases le permitirán elegir su respuesta de entre un banco de acciones posibles. Pero si el sistema no logra identifi-

car con claridad los gestos y llegar al rango impuesto por la programación, el sistema focalizará su atención en una escucha de la textura global del ensamble. Esta misma forma de escucha textural también puede ser clasificada en escenas pre-programadas usando un modelado de sistemas dinámicos, implementado por Van Nort. De manera que FILTER puede discernir entre distintos tipos de texturas musicales, y de igual manera asignar un grado de certeza al identificar texturas arquetípicas almacenadas en su memoria. Esta memoria considerada como episódica usa el procedimiento de Factor Oracle que anteriormente fue usado en OMax y posteriormente en otro sistema llamado MIMI.

FILTER reacciona partiendo de distinguir material muy similar o muy diferente de acuerdo con los materiales almacenados en su memoria, es decir, sus reacciones están reguladas por los gestos relativamente más sobresalientes del contexto, a lo cual corresponden acciones asignadas de acuerdo al modo de escucha del sistema. Asimismo, FILTER puede hacer estiramientos temporales y transposiciones, con lo cual puede tener un abanico muy amplio de recombinatoria para interactuar en la improvisación.

> En esta implementación, un conjunto de miembros de la población de N-dimensiones se mueven dentro de un complejo simplicial *(simplicial complex)*[6] donde cada nodo está asociado con un estado de comportamiento del sistema. El miembro de la población asociado con el estado gestual/-textural actualmente más destacado se utiliza para interpolar los nodos del complejo simple que lo rodea, determinando un estado de comportamiento del sistema. Esto puede considerarse una nube de estados posibles que se mueven con una naturaleza casi física tal como lo determina la salida del módulo de escucha.

Las acciones que el sistema puede realizar dependiendo de lo que escucha son las siguientes:

[6]En matemáticas un complejo simplicial es un conjunto de lineas, segmentos, triángulos y su contraparte de N-dimensiones.

- **Cualidad rítmica** mayor probabilidad de que los fragmentos reconocidos sean repetidos, así como variaciones sobre la repetición.

- **Cualidad salvaje** aumenta la probabilidad de que el sistema imite al improvisador, usando material reciente y pasado de forma caótica.

- **Estabilidad** mayor probabilidad de que el sistema cambie súbitamente sus estados de comportamiento.

- **Sostenimiento (sustain)** favorece los sonidos tenidos.

- **Densidad** dependiendo la densidad el sistema cambia entre duración y mayor o menor espaciamiento en las frases de salida.

Por último, FILTER es un antecedente directo del proyecto planteado en esta tesis por varias razones: en términos sonoros hay una empatía estética debido a su búsqueda dentro de la música libremente improvisada, particularmente desde la noción de escucha profunda planteada por Pauline Oliveros. También es significativa la importancia que tiene FILTER en términos técnicos ya que su sistema de discernimiento gestual basado en el aprendizaje supervisado resultó útil para una de las aproximaciones que planteo para el reconocimiento de timbres en la libre improvisación. Por sus características y además por los resultados sonoros, FILTER fue una una gran referencia, ya que me permitió seguir trabajando con el sistema de escucha, perfeccionarlo y plantear las formas interactividad del sistema planteado en esta tesis.

1.1.4. Clara

Clara es un sistema basado en redes neuronales recurrentes LSTM, Long Short Term Memory por sus siglas en ingles[7] para la composición de musica de cámara polifónica. fue desarrollado bajo los entornos de programación Pytorch, music21 y la librería FastAI. Christine McLeavey Payne, pianista y desarrolladora del proyecto, usa un rango de notas restringido para poder generar mayor redundancia en los datos. En vez de utilizar las 88 teclas del piano reduce el rango de posibilidades a notas

[7] Éstas serán descritas en el capítulo 4.

de 62, comprimiendo la información que pudiera estar por encima de la tecla 62. Al olvidarse de los extremos agudos, que finalmente no son muy recurrentes en la música que quiere analizar, es más factible que el proceso de entrenamiento pueda llegar a predicciones más adecuadas sobre datos nunca antes vistos. El código del proyecto lo mantiene abierto, a través de la plataforma github y es posible descargarlo desde el siguiente enlace.[8] Una vez descargado es posible hacer nuevas versiones musicales con diversos datos de entrada utilizando el modelo que previamente ella entrenó con miles de archivos MIDI de musica clásica.

Para poder hacer este análisis, la autora codificó los archivos MIDI en archivos de texto destacando elementos como el inicio y el final de cada nota, el volumen, la información del instrumento que toca y las marcas de tempo de la pieza. Estos elementos pueden cambiar arbitrariamente además de la cantidad de notas tocadas. Para la subdivisión rítmica hizo un muestreo de 12 o 4 partes por cada cuarto de nota, esto con el fin de poder detectar muchas más subdivisiones como tresillos o dieciseisavos, en el caso de 4 partes los tresillos podrían variar ligeramente pero mientras menor información haya será más sencillo para el modelo ser entrenado y llegar al punto convergencia donde ya no puede aprender más de los datos presentados. Además esta frecuencia del muestreo puede ser modificada para los fines necesarios. A continuación presento una forma en la que la autora representa los datos MIDI a texto:

```
1 bach piano_strings start tempo90 piano:v72:G1 piano:v72:G2 piano:v72:
    B4 piano:v72:D4 violin:v80:G4 piano:v72:G4 piano:v72:B5 piano:v72:
    D5 wait:12 piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5 wait
    :4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4 violin:v0:G4
    piano:v0:G4 wait:1 piano:v72:G5 wait:12 piano:v0:G5 wait:5 piano:
    v72:D5 wait:12 piano:v0:D5 wait:5 piano:v72:B5 wait:12
```

Sin bien Clara no es un desarrollo que esté enfocado en la predicción de estructuras musicales, es interesante analizar lo que es capaz de generar en términos formales dada una serie de datos iniciales. Partiendo del análisis de secuencias musicales, este tipo de algoritmos son capaces de inferir otros niveles musicales que no fueron explícitamente programados para ello. Dependiendo de su programación, pueden inferir

[8] https://github.com/mcleavey/musical-neural-net

armonía, ritmo, densidad, estilo y forma, aunque esta última de una manera no muy detallada.

La autora creó un modelo de evaluación que trata de identificar si la música generada fue compuesta por un humano o por redes neuronales. Esta aproximación le permitió evaluar que tan bien o qué tan mal el modelo puede generar versiones y obtener las mejores versiones generadas por el modelo. Otro método que utilizó para evaluar los resultados, fue recurrir a la *prueba de Turing*. En este ejercicio presenta dos grabaciones de audio, una es la real correspondiente a los archivos MIDI creados por diferentes compositores y la otra es una composición generada por la red neuronal recurrente. La idea es que un humano valide cual de las dos grabaciones es la original, a partir de esto puede guardar los datos en una base de datos y evaluar la generación del modelo de manera favorable o desfavorable.

> Los modelos obtienen un puntaje alto cuando cada generación individual se parece mucho a un compositor humano específico, y cuando, en promedio, hay generaciones que coinciden con todos los diferentes compositores humanos. Un modelo obtendría una puntuación baja por generar solo piezas como Chopin y nunca generar piezas como cualquiera de los otros humanos. Alternativamente, es interesante combinar estos dos puntajes de modelo. Se podría crear un estilo musical novedoso encontrando muestras que el crítico considera reales", pero que el clasificador de compositores no coincide con ningún compositor humano.

Si bien con la debida atención a los fragmentos musicales que propone McLeavey es posible determinar si Clara o un humano los compusieron, el sistema puede hacer una propuesta musical que tal vez para una escucha poco entrenada resulte difícil detectar quien realmente compuso los pequeños fragmentos musicales, considero que hay ciertos indicios que nos pueden llevar a detectar al agente artificial. El primer indicio que detecté, fue la aproximación rítmica de Clara; por momentos puede ser un poco burda, poco cuidada o repetitiva, en algunos casos tendiendo a algún tipo de glitch. Otro elemento que detecté, fue la repetición constante e incisiva de algunas notas. Creo que sería poco común encontrar este tipo de patrones en composiciones

del estilo, escuchar cuatro veces una misma nota repetida en varias ocasiones fue una buena señal. En el siguiente enlace se encuentra la encuesta para escuchar y evaluar los resultados de Clara frente a los de un humano: `http://christinemcleavey.com/human-or-ai/`

Clara es un proyecto ligado a el proyecto *Magenta* de *Google* y *MuseNet*[9] el cual fue creado a partir de redes neuronales recurrentes y que es capaz de generar composiciones musicales en diferentes estilos con diferentes instrumentos musicales. Este sistema fue desarrollado desde una perspectiva que les permitiera a los autores no enseñarle específicamente sobre armonía, contrapunto y ritmo o estilo sino que a través del entrenamiento con varios miles de archivos MIDI el sistema pudiera inferir de manera automática que tipo de secuencias corresponden a una secuencia ya dada. Asimismo, MuseNet esta basado en un sistema hermano el GPT-2 y GPT-3, el cual emplea aprendizaje sin supervisión para predecir nuevas estructuras de texto.

Son interesantes pero también controversiales las formas en que muchos sistemas autónomos se insertan en la industria musical digital, la poca crítica que hay alrededor de su desarrollo y hacia su utilización, denotan aún más, el fuerte trabajo reflexivo y crítico que hace falta generar en torno a su producción y consumo. Aún así, estos sistemas son poco a poco más socorridos por las plataformas digitales y los estándares de la tecnología digital en la industria discográfica, llenando de contenidos genéricos, creados por sistemas autónomos; la oferta artística y musical a través de la red. Incluso en algunos casos sin comercializarse como producciones hechas por IA, sino pasando por artistas o productores humanos.[10] [11] [12]

En abril del año 2019, Magenta ofreció un concierto en directo transmitido en linea que hasta el momento ha alcanzado una difusión de casi 500k espectadores. Este performance estuvo lleno obras en distintos estilos todas ellas estrenos mundiales que incluso los mismos autores no habían escuchado antes. [13]

[9] `https://openai.com/blog/musenet/`

[10] Para mayor información sobre este tema, revisar mi tesis de maestría *Hacia una escucha automática de la espontaneidad: relaciones complejas entre la libre improvisación, la escucha y los sistemas de aprendizaje automático.*(Escobar Castañeda, 2018) en el apartado: *Implicaciones sobre el uso del las tecnologıéas de aprendizaje y la escucha automático*

[11] `http://www.flow-machines.com/ai-music/` Fecha de consulta 7 de febrero 2018

[12] `http://www.flow-machines.com/ai-makes-pop-music/brero2018`

[13] `https://bit.ly/3TBjUS8`

Si bien, Clara o MuseNet son proyectos interesantes debido a sus múltiples matices en cuanto a sus métodos para procesar la información MIDI, ambos tienen sus límites, ya que no contemplan una interacción humano-máquina, ni tampoco el análisis de señales digitales de audio. Esto supone un cambio sustancial en las técnicas/aproximaciones y el grado de complejidad que la resolución de un problema como el que planteo en esta investigación implica. Por ello, no seguiré desarrollando más detalles de este proyecto. Para una mayor información sobre este proyecto recomiendo consultar la entrevista que le hacen a McLeavey sobre Clara. [14]

Hago mención a estos desarrollos tecnológicos debido a que las implementaciones a las que recurren para la predicción de secuencias temporales en la evolución musical son prácticamente las mismas al proyecto que planteo en esta investigación. Con sus matices y arquitecturas particulares, ambos proyectos recurren a las redes neuronales recurrentes específicamente las redes LSTM. Otro elemento que llama mi atención es el de la estructura en donde a partir del conocimiento individual de las secuencias temporales de un corpus de obras musicales es posible que el sistema pueda reconstruir estructuras sintagmáticas de nivel superior y dar la ilusión de que sabe como proceder en términos formales en el desarrollo de cualquiera de los estilos que trata de imitar. Asimismo, estos proyectos fueron de gran ayuda para el construir el imaginario y conceptualización algorítmica de SEALI.

1.1.5. Piano + AI

Un nuevo instrumento AI entrenado con técnicas de piano

Un proyecto muy interesante, aunque actualmente no tiene mucha documentación salvo una grabación y lo que he podido platicar con uno de sus creadores, es Piano + AI. Este sistema fue desarrollado en 2021 por Iván Paz, Philippe Salembier, Josep Maria Comajuncosas y Joan Canyelles en colaboración con el pianista Marco Mezquida. El sistema que utilizan es un instrumento que emplea la inteligencia artificial para responder e interactuar a través de la escucha de algunas técnicas pianísticas. Piano + AI es un proyecto interdisciplinario que combina especialistas en inteligencia artificial y el procesamiento de señales digitales de audio y músicos,

[14] `https://blog.floydhub.com/generating-classical-music-with-neural-networks/`

vinculados con la Universidad Politécnica de Catalunya (UPC) y con de la Escuela Superior de Música de Catalunya (ESMUC). El trabajo se dividió en tres partes; Iván Paz y Philippe Salembier se enfocaron en la parte metodológica e implementación de algoritmos de redes neuronales artificiales y el procesamiento de las señales digitales de audio; Josep Maria Comajuncosas y Joan Canyelles se enfocaron en el diseño de los paisajes sonoros sintéticos; y Mario Mezquida y Anna Francesch en las grabaciones de los materiales para entrenar al modelo y la improvisación con la IA. [15]

Este sistema de inteligencia artificial responde a partir de diferentes paisajes sonoros sintéticos, como sus autores los han llamado, generados a partir del lenguaje de programación Supercollider, librerías como ML.Star de Max/msp, on-line leaning, Real Time Composition Library[16] y algoritmos de reducción de dimensiones como UMAP[17]. Esta interacción genera una estructura colaborativa donde el proceso de retroalimentación que involucra al pianista y la inteligencia artificial juega un papel constructivo y desarrollo estructural importante. En la siguiente nota al pie, es posible apreciar uno de los performances que este grupo de colaboradores Ha realizado en el marco del festival SONAR S+T+ARTS.[18]

En este performance es evidente que el mismo desarrollo formal y estilístico que el pianista propone, va guiando la improvisación a diferentes estados sonoros acústicos. A través de estos, la IA avanza a manera de activación de diferentes "patches" o configuraciones sonoras que varían su densidad y frecuencia de aparición, en función de los descriptores de audio analizados y de las clasificaciones que el modelo de aprendizaje devuelve. Esto es apreciable desde la primer sección, donde sonidos similares a una síntesis granular o sonidos procesados mediante *delays* en *crescendo*, acompañan los sonidos que Marco Mezquida realiza dentro de la caja del piano, rasgando sutilmente las cuerdas y gradualmente transitando hacia una técnica tecladística. Mientras tanto, el sistema sonoro sigue recurriendo a los sonidos granulados pero solamente varia su frecuencia de aparición. Esto hace notar inmediatamente que el sistema no responde claramente a los elementos sonoros presentados sino que parece ignorar el

[15] `https://bit.ly/3dhAnuj`
[16] `http://www.essl.at/works/rtc.html`
[17] `https://umap-learn.readthedocs.io/en/latest/how_umap_works.html`
[18] `https://bit.ly/3eP2fGB`

tipo de materiales que el pianista propone a nivel de densidad y tímbrica. Después, súbitamente el sistema sonoro deja de producir los sonidos granulados y comienza a sumarse a la masa sonora del pianista, haciendo algunas texturas sutiles y produciendo esporádicamente sonidos en un registro agudo. Esto me lleva a pensar que ambas perspectivas podrían parecer independientes de sí mismas y sustentarse individualmente. Al escuchar todo el performance, percibo que la IA no colabora al mismo nivel que el pianista, en este sentido no se siente una interacción puntual ni clara por parte de la IA, al punto en el que la improvisación pareciera haber sido la misma sin la colaboración humano-máquina. Aunque hubieron algunas excepciones donde es perceptible cierta cohesión sonora, más bien parece producto de las coincidencias afortunadas del azar y coincidencia emergente. Es cierto que fue un performance muy interesante y que disfruté enormemente al escucharlo dada la maestría de Mozquida, pero desde lo que escucho, que es la evidencia tangible a la que puedo acceder dada la falta de documentación de este proyecto, considero que hay mucho trabajo por hacer en términos de interacción humano-máquina.

1.1.6. Apperceptions

Musica para improvisador humano con improvisador computacional

En 2020 el compositor Taylor Brook desarrolla *Scuffed Computer Improviser (SCI)*, un patch de Max/MSP programado bajo las librerías *ml.star* y *ml.markov* de Ben Smith. Ambas librerías están especializadas en el aprendizaje automático en linea y la interacción musical en tiempo–real. Al igual que FILTER y Omax, SCI es un sistema basado en un corpus de audio generado "al vuelo", por lo que puede improvisar en vivo con cualquier entrada de audio. De este modo, su interacción se abre y limita, al mismo tiempo, a la instrumentación o la tímbrica con la que se está improvisando. Además, puede interactuar a través de diversos modos de comportamiento, estos modos tienen la posibilidad de: duplicar, seguir, empatar, invertir, liderar, imitar (usando cadenas de markov) y reaccionar (de modo reactivo) al audio de entrada. Aunado a ello, estos comportamientos pueden ser automatizados y controlados por el patch a partir de los datos de entrenamiento del modelo. Otra característica de SCI es que fue programado para entregar el audio en mono, estero, y en formato cuadrafónico y hexafónico, agregando mayor versatilidad a su interactividad.

Dentro del patch de Max/MSP hay un botón con la leyenda *train* o aprendizaje, al presionarlo, SCI comienza a analizar el flujo de audio y lo deposita en una pileta (*pool*) que posteriormente genera una base de datos de sonidos los cuales puede reproducir y cortar de acuerdo con su programación. SCI analiza altura, ritmo, brillo, amplitud y planitud espectral (o armonicidad, es decir, qué tan rugoso o plano es un sonido). Después de que el sistema ha sido entrenado, para comenzar con la automatización, es posible activar o desactivar el improvisador de SCI mediante el botón *On/Off*. A partir de este momento el improvisador comienza a reaccionar al audio de entrada, cambiando entre los diversos modos de comportamiento de acuerdo con lo que escucha al momento. Si el botón *On/Off* se desactiva, los comportamientos pueden ser seleccionados de forma manual. SCI cuenta con 4 improvisadores que pueden ser entrenados en diversos momentos a lo largo del performance, activados de manera individual o simultanea. Una función interesante es que cuenta con un sistema de retroalimentación que puede alimentar a cada uno de los improvisadores de regreso, de manera que cada improvisador de SCI puede aprender nuevamente de las improvisaciones de los otros agentes.

En los siguientes enlaces una pequeña documentación al proyecto, el patch de Max/MSP, además de algunos tutoriales donde Taylor Brook explica someramente el funcionamiento de SCI.[19] [20]

Derivado de este sistema Taylor Brook realizó *Apperceptions: Musica para improvisador humano con improvisador computacional*, este album es un dueto entre SCI y el propio Brook donde la computadora es entrenada al vuelo para detectar y reaccionar a los sonidos de entrada de él mismo. "Este proceso se puede escuchar en la música ya que cada pista comienza con un solo que le enseña al improvisador cómo ser musical."[21] Algo interesante en la conceptualización de este disco es que la guitarra eléctrica tiene una afinación justa diferente en cada una de las improvisaciones, de manera que la progresión que propone una improvisación a otra, representa un movimiento a través del espacio armónico de intervalos que están estrechamente relacionados. Al respecto Brook comenta sobre su sistema lo siguiente:

[19] `https://taylorbrook.bandcamp.com/album/apperceptions`
[20] `https://www.taylorbrook.info/sci`
[21] `https://taylorbrook.bandcamp.com/album/apperceptionsConsultadoel1deDiciembrede2021.`

> El "improvisador de computadora desgastado" es tosco alrededor de los bordes, la imperfección y la falta de claridad del sistema de inteligencia artificial es una característica del software, diseñado para empujarme mientras improviso como haría un compañero.

Conclusión de capítulo

Pese a que partí de la premisa de presentar proyectos que usaran tecnologías como el aprendizaje y la escucha de máquina en el cruce de la práctica de la improvisación libre, concentrándose en sus exploraciones tímbricas y con un particular interés en el análisis de la estructura musical de esta práctica, inevitablemente esta especificidad me llevo por otro camino. La falta de proyectos que aborden específicamente el cruce entre estos temas me llevó a explorar proyectos que develan algunas pistas que sirvieron de guía e inspiración para los capítulos posteriores de esta tesis.

Seguramente estaré dejando muchos proyectos fuera de este pequeño pero necesario estado del arte, lo que intenté, dado el panorama de este rastreo, fue presentar los proyectos más relevantes en términos de las herramientas, técnicas y metodologías empleadas para conocer las exploraciones que otros creadores musicales y programadores han tenido respecto al desarrollo de sistemas inteligentes interactivos para entablar diálogos, primordialmente, humano-máquina. Por ejemplo, aunque en el caso de Clara, no precisamente exploran el análisis de señales de audio y menos aún la improvisación libre, la aproximación del análisis estructural es similar a la empleada en el desarrollo del sistema propuesto en esta tesis. En el caso de Piano + AI, que no cuenta con mucha documentación, resulta sumamente relevante su mención tomando en cuenta el resultado estético al que llegan dentro del contexto en el que se inserta este performance.

Si bien son muchas las posibilidades creativas y el camino que abren estos proyectos es muy amplio[22], es necesario preguntar porqué usar éstas tecnologías, porqué desde ahí, porqué ahora y a quién le interesa el despliegue masivo y el posicionamiento en la utilización de la inteligencia artificial en la música; como lo hemos visto

[22]Basta conocer algunos de los proyectos de los artistas invitados en festivales como SONAR 2021-2022 `https://bit.ly/3zYghgI`

cada vez más en los festivales actuales de música electrónica. Evidentemente no solo se limita a este espectro sino cada vez en más en ámbitos creativos, dedicados a la productividad, la gestión y el estilo de vida. Desde éstas preguntas es posible abordar el tema de manera profunda, cuestionar los despliegues y la velocidad que tienen e indagar sobre las propias formas de hacer y pensar estas tecnologías en el cruce de los paradigmas actuales en torno a la creación artística. Por ello considero necesario poder conocer las herramientas y los métodos dominantes para no solo déjanos llevar por el flujo de esa velocidad sino poner cierta resistencia y partir desde una mirada crítica hacia la industria positivista que tiende hacia cientifización del arte y el auge que actualmente surge entre una multitud de artistas que se encuentran en el cruce del arte y la utilización generalizada de la inteligencia artificial.

Capítulo 2

Escuchar la impermanencia creativa

Antes de profundizar en la parte técnica de la investigación, este capítulo tiene el objetivo de investigar algunos de los aspectos que considero relevantes tanto a nivel histórico como sonoro y estructural sobre la improvisación libre. Abordarlos me permite observar que las discusiones sociales en torno a la improvisación libre y el free jazz, promueven el surgimiento de múltiples perspectivas creativas comprometidas con la búsqueda de otras formas de hacer música deviniendo así sus propias aproximaciones estéticas características y sus medios para producirse. Ello me permite conocer las implicaciones que conlleva el aproximarse a la improvisación libre para acercarme, con un mayor conocimiento/(compromiso), a el desarrollo algorítmico que acompaña la investigación. Además, este abordaje me permite indagar el porqué me resulta tan interesante partir de esta forma particular de hacer música en mi quehacer creativo. Asimismo, presento los aspectos conceptuales que implicaron la creación de SEALI, el cual fue desarrollado a la luz de tres perspectivas:

1. El **ciclo de retroalimentación práctica artística, investigación y desarrollo tecnológico** que es deviene en un bucle constante de afectaciones a cada uno de los aspectos que componen este ciclo y los objetivos iniciales de la investigación. Para ello parto del concepto de la *mediación técnica* de Bruno Latour (Latour,

2001), quien lo define como los múltiples procesos de afectación entre humanos y máquinas que permiten la actualización, mutación y *rodeos* producidos en los objetivos iniciales de un problema que involucra la interacción entre actores humanos y no humanos.

2. La fenomenológica, entendida desde la **escucha profunda** de Pauline Oliveros (Oliveros, 2005), quien la define como un proceso de atención hacía múltiples niveles de información que nos permite expandir la percepción auditiva para entrelazarla al proceso complejo presente en el continuo del espacio/tiempo sonoro.

3. La **agencialidad algorítmica** o **algoritmicidad**, definida como la capacidad que tienen los algoritmos para establecer procesos de intercambio y afectación con su entorno tanto dentro como fuera de una computadora (Rutz, 2016).

2.1. La improvisación libre

Si bien es difícil hablar de un momento exacto sobre el surgimiento de la improvisación libre, ya que se deriva de muchos ámbitos musicales como el free jazz o de algunas corrientes académicas del *avant garde* musical, más aún, teniendo en cuenta que algunos improvisadores han llegado a considerar que la libre improvisación fue la primer música que crearon los humanos, podríamos decir que la práctica como la conocemos actualmente surge en los años 60s en Europa y Estados Unidos.[1]

> Históricamente, es anterior a cualquier otra música (la primera interpretación musical de la humanidad no pudo haber sido otra cosa que una improvisación libre) y creo que es una especulación razonable que, en la mayoría de los casos desde entonces, habrá habido alguna creación musical, descrita más acertadamente como improvisación libre. (Bailey, 1980)

Para poder hablar de su genealogía es importante hablar del free jazz, el cual, encabezado por músicos como Cecil Taylor, Ornette Coleman, John Coltrane, Albert Ayler, Eric Dolphy, Alvin Fielder, Archie Shepp, entre otros, buscaban abrir un espacio político de expresión a favor de las luchas por los derechos civiles, posicionándose

[1] urlhttp://www.paristransatlantic.com/magazine/interviews/rowe.html

en contra del linchamiento y la encarcelación de lideres políticos, cuestionando las precarias condiciones de vida de la época, así como las estructuras de poder en la industria musical, con el objetivo de buscar mejores condiciones socio-económicas. Al respecto George E. Lewis escribió:

> Los protagonistas [del free jazz] declaran repetidamente que su trabajo representaba una respuesta personal a las condiciones sociales y políticas, así como a las condiciones económicas en las que trabajaban regularmente. Según Shepp, por ejemplo, la condición compartida de aparente servidumbre económica de los músicos fue un factor primordial en su insatisfacción. Hablando de las estructuras dominadas por los blancos de la industria de la música jazz en términos que recuerdan claramente la esclavitud, Shepp declaró que "tú eres el dueño de la música, nosotros la hacemos. Por definición, eres dueño de las personas que hacen la música. Posees en nosotros trozos enteros de carne. (Lewis, 2008)

Desde la búsqueda por cuestionar el sistema imperante de la época, muchos músicos comenzaron a integrar prácticas colaborativas de creación musical, empleando elementos performáticos y recursos sonoros que permitieron una expansión musical en un terreno fértil pero escabroso. Esto produjo la ruptura de las jerarquías compositor-intérprete, los modos de producción musical dominantes, la tonalidad, la modalidad y los ritmos estables, para conjugar otras técnicas instrumentales, expresivas y aproximaciones polirrítmicas. Además, se exploró la fabricación de nuevos instrumentos[2], dinámicas de interacción entre los músicos y exploraciones que indagan sobre el proceso creativo desde el contenido espectral, los ritmos intrincados, la dupla entre novedad/monotonía, el juego con la energía y la densidad espectral. La conjugación que se forma de todos estos elementos y agentes abre la posibilidad de explorar los límites de cada uno, dando lugar a un conjunto entrelazado de intersticios que ocasionan la libre improvisación. Así, los múltiples agentes implicados (humanos, técnicas, diálogos, espacios, tiempos y materias), coexistiendo, afectando y siendo

[2]Un ejemplo es el *daxophon* de Hans Reichel. `https://youtu.be/A8uGNY2Qf9Y`

afectados por otros agentes en sus intra-acciones mutuas, posibilitan el surgimiento de esta praxis.[3]

Es importante señalar que entre la improvisación libre y el free jazz hay una gran diferencia en su génesis, lejos de apuntar a los intereses políticos de éste último, la improvisación libre, o como la llama Derek Bailey improvisación no idiomática (Bailey, 1980), se posiciona como una contrarespuesta a los avances musicales producidos en las esferas hegemónicas de la música, particularmente en el jazz o el *avant garde* de la música académica. Con el paso de los años ambas expresiones musicales logran delimitarse en una corriente estilística particular donde prácticas, crítica y públicos dan lugar y cohesionan su espacio de posibilidad.

> Para complicar el panorama, estaba el hecho de que, a partir de la década de 1950, el jazz estadounidense negro, en particular el bebop, estaba ganando aceptación como una forma de música artística. Como tal, esta nueva música se encontró compitiendo con los productos sonoros históricamente certificados de Europa, y las respuestas a esta nueva competencia, articulada desde suelo europeo, fueron a veces tan extremas como lo habían sido en los Estados Unidos en los años cuarenta y cincuenta. [...]
>
> [...] John Coltrane, conocido por sus suaves y expansivas declaraciones de estética y propósito "John Coltrane y Eric Dolphy responden a los críticos de jazz". En lugar de ofrecer anécdotas sobre la vida en la carretera y listas de colegas que eran todos "buenos músicos", Dolphy y Coltrane, en un tono bastante prosaico, destacaron cuestiones de infraestructura, estética, técnica, contenido y forma en su trabajo. (Lewis, 2008)

[3]"En mi explicación del realismo agencial, todos los cuerpos, no solo los cuerpos humanos, cobran importancia a través de la performatividad del mundo, su intra-actividad iterativa. La materia no se concibe como un mero efecto o producto de las prácticas discursivas, sino como un factor agencial en su materialización iterativa y, la identidad y la diferencia se reelaboran radicalmente. En particular, [...] lo que comúnmente tomamos como entidades individuales, no son objetos determinantemente separados ni posesiones [de algo o de alguien], sino más bien («partes de») fenómenos enredados (intra-acciones materiales-discursivas) que se extienden a través de (lo que comúnmente toman como lugares y momentos separados en) el espacio y el tiempo (donde las nociones de "lo material" y "lo discursivo" y las relaciones entre ellos se desatan de sus fundamentos (anti)humanistas y se reelaboran)."(Barad, 2011)

Ejemplo de esta crítica parte de la esfera hegemónica de la música académica contemporánea y el escándalo que suscitó el que Jean Luc Ferrari quien dirigiera la serie de documentales para la televisión francesa *Les Grandes Repetitions*, sobre compositores contemporáneos de la talla de Edgar Varese, Karlheinz Stockhausen, Oliver Messiaen y Hermann Scherchen e integrara en dicha serie a Cecil Taylor. [4]

> "Al menos", escribió, "este brebaje de brujas es el producto final lógico de una filosofía en bancarrota de ultraindividualismo en la música. ¿Improvisación colectiva? Tonterías. La única apariencia de colectividad radica en el hecho de que estos ocho nihilistas se reunieron en un estudio a la vez y con una causa común: destruir la música que les dio nacimiento". (Lewis, 2008)

Este panorama fue paulatinamente definiendo a lo largo de los años, la improvisación libre la cual sigue partiendo de elementos musicales muy específicos, definibles y categorizables, que incluso puede ser confundida con otras tradiciones musicales como la música *avant garde* o experimental. (Bailey, 1980)

> Es cierto que muy a menudo se agrupan, pero esto probablemente se hace en beneficio de los promotores que necesitan saber que lo único que tienen en común es una incapacidad compartida para captar la atención de grandes grupos de oyentes ocasionales. Pero aunque puedan compartir el mismo rincón del mercado, son fundamentalmente muy diferentes entre sí. Los improvisadores pueden realizar experimentos ocasionales, pero creo que muy pocos consideran que su trabajo sea experimental. [...] Hay innovaciones hechas, como era de esperar, a través de la improvisación, pero el deseo de mantenerse a la vanguardia no es común entre los improvisadores. Y en cuanto al método, el improvisador emplea lo más antiguo de la música. (Bailey, 1980)

A lo largo de la historia de la improvisación, ésta no siempre ha sido bien recibida por muchas audiencias debido a que el hecho de la inmediatez de su producción puede

[4] `https://www.youtube.com/watch?v=Nb8-Pfxu4Og&ab_channel=SOUNDOHM`

conducir fácilmente a considerarla como una música de menor valor. En la búsqueda por desmitificar estas apreciaciones que algunos hemos tenido con ciertos productos musicales, Clément Canonne hace un análisis muy interesante en su texto *escuchando la improvisación* (Canonne, 2019), donde a través de un estudio psicoacústico, hace un recuento sobre qué diferencias habría al escuchar música improvisada y una pieza compuesta y, desde ahí, cómo el hecho de saber si lo que escuchamos es música improvisada o no, afecta nuestra experiencia de escucha y su valoración per se. En su texto menciona que:

> [...] existe la posibilidad de que percibir una pieza musical como una improvisación no sea de ninguna manera diferente a percibir exactamente la misma pieza como la interpretación de una composición preexistente. Si lo que es estéticamente valioso en una pieza musical puede detectarse simplemente escuchándolo, en lo que realmente escuchamos, entonces saber que una pieza musical es improvisada o compuesta no debería hacer ninguna diferencia. (Canonne, 2019)

Esta práctica musical parte de una genealogía representada por un grupo de músicos mayoritariamente europeos que buscaba crear sus propios lenguajes musicales más allá de la imperante composición *avant garde* de la época y las tendencias estilísticas del free jazz. Algunos de los representantes más destacados serían Derek Bailey, Evan Parker, Eddie Prévost, Keith Rowe, John Tilburi, Anthony Braxton, Peter Brötzmann, John Zorn, Pauline Oliveros, George Lewis, Henry Kaiser, Fred Frith, Ennio Morricone, entre muchos otros. Además, entre los grupos de improvisación más destacados podemos encontrar a el *Ensamble de Música Espontánea*, la *Compañía de Improvisación Musical*, el grupo *Gruppo Nuova Consonanza*[5], *Musica Elettronica Viva* (MEV), el *ensamble de arte de Chicago*, *AMM* y *AACM* (Asociación para el Avance de los Músicos Creativos).

Bailey menciona que las características de la libre improvisación las establece la identidad musical y sonora de las personas que la hacen.(Bailey, 1980) Esto abre muchas posibilidades para la improvisación, sin embargo, con el paso de los años podemos escuchar que sus características estéticas la insertan cada vez más como un

[5]https://bit.ly/3hvvcsd

género musical consolidado, de este modo también sería posible hablar en términos de tradición dentro de la improvisación libre, comenzando con el estilo de Bailey el cual sigue siendo un referente para muchos improvisadores.[6][7] Si bien, no necesariamente lo que hacia el Art Ensemble of Chicago en los 60 [8] se parece a lo que actualmente hacen los improvisadores libres, desde mi perspectiva hay una fuerte tendencia estilística que se se ha consolidado, con el paso de los años en la improvisación libre. En los siguientes dos ejemplos es posible escuchar estas equivalencias a nivel sonoro pese a los ya más de 50 años de diferencia:

AMM - AMMusic 1966

`https://www.youtube.com/watch?v=BwgkBZ-FLW0&ab_channel=TheAvantgardeCave`

Milo Tamez una semana de bondad: sesioén V

`https://www.youtube.com/watch?v=6lGfEFdHcx8&ab_channel=MiloTamez`

Desde esta perspectiva, un elemento destacable sería "el aura de la improvisación libre", esa sustancia que viaja, dialoga y se construye, a lo largo del tiempo, bajo supuestos acuerdos, conscientes o no, sobre qué se espera abordar en la creación de una libre improvisación. Tan es así, que la misma escena, improvisadores y audiencia, sabe bien al ir a un concierto de libre improvisación, qué se espera escuchar, a qué prestarle atención y cuáles son los códigos de conducta en un concierto de improvisación libre, donde generalmente prima un respetuoso y contemplativo silencio que posibilita la misma exploración de éste como un elemento constructivo sumamente importante, que va tejiendo la expectativa en torno a la improvisación, además de una amplia gama de umbrales dinámicos que habilitan el pensamiento sónico de cada improvisador en resonancia con su entorno. A este respecto encuentro sumamente relevante la siguiente reflexión de Jean-Luc Nancy:

> Debemos abrirnos a la resonancia del ser o al ser como resonancia. El silencio aquí debe de entenderse no como una privación, sino como una disposición a la resonancia: como en una situación de silencio perfecto, en

[6] `https://www.youtube.com/watch?v=A5dz_1meBjY&ab_channel=donjuanauxenfers`
[7] `https://www.youtube.com/watch?v=vBnZDhgdX9Q&ab_channel=0ariyle0`
[8] `https://www.youtube.com/watch?v=xLd-VybGzsA&ab_channel=JacksonBrown`

> donde podemos escuchar resonar nuestro propio cuerpo, nuestro aliento, nuestro corazón y toda su caverna estrepitosa. (Nancy, 2008)

Si en el free jazz había una clara búsqueda por la libertad en términos socio-económicos produciendo la ruptura con los estándares de su práctica, de dónde viene la idea de "lo libre" en la libre improvisación. Desde el análisis que he realizado y el acercamiento que tengo hacía la improvisación libre, puedo decir que "lo libre" ocurre en muchos aspectos de esta práctica. No solo en las formas estructurales que produce, que, si bien, al momento de improvisar son susceptibles a modificaciones, son maleables y abiertas al cambio en todo momento de acuerdo con el desarrollo sonoro contextual, sino que el concepto de "lo libre" también ocurre en otros aspectos que suceden al materializar esta práctica. Algunos de éstos, giran en torno a los "haceres improvisativos" que generalmente desbordan la propia práctica performativa. Éstos aspectos involucran la gestión, muchas veces autónoma, de espacios privados que dan lugar a los performances, en otros casos, a través de instituciones públicas, pero casi siempre promovidos por los mismos improvisadores de la escena local. Es importante destacar que en esta práctica musical, en general, la falta de patrocinios e intereses por una cultura *mainstream*, permiten observar un interés genuino por parte de los improvisadores para generar estos espacios de encuentro. Ello permite crear el foro y la plataforma de comunicación entre improvisadores y una audiencia muy diversa, que al igual que la propia práctica, también está en constante cambio y apertura hacia nuevas expresiones sonoras.

En este punto, me gustaría hablar de la escena musical de la libre improvisación de la que pude ser parte en la Ciudad de México de 2012 hasta antes del comienzo de la pandemia a inicios del 2020. En esta escena, los foros y espacios para tocar, en muchas ocasiones eran casas privadas, departamentos, azoteas y bares rentados que a través del tiempo se fueron haciendo populares entre la comunidad, los cuales, permitieron visibilizar un gran abanico de propuestas improvisativas, tanto, locales, nacionales y extranjeras. Así, estos espacios de encuentro permitieron desplegar una amplia gama propuestas estéticas, así como dar lugar a la compartición de espacios de creación en torno a la libre improvisación musical tanto a músicos/no músicos y público de diversas latitudes.

Dentro del espacio contextual de esta práctica y todo lo que ella implica, surge la figura del improvisador como agente que desempeña varios roles en momentos distintos ("haceres improvisativos"). Cabe señalar que esta multiplicidad de roles no se limita a la escena de la música improvisada, pero particularmente veo de manera más presente los procesos de autoorganización en esta práctica más que en otras. Dentro de estos haceres improvisativos, el improvisador por momentos puede ser gestor de conciertos, facilitador de lugares de encuentro, ingeniero de audio de los performances, recopilador de improvisaciones para posteriormente lanzar un álbum a la venta; ya sea a través de cassette, disco compacto, en vinilo, en plataformas digitales o la conjunción de estos soportes de audio para compartir la música; por si esto no fuera poco, en otros momentos, el improvisador también es público y escucha.

Por otro lado, en comparación con otras prácticas musicales improvisativas (o improvisaciones idiomáticas (Bailey, 1980)), es importante notar que en la improvisación libre no hay una clara distribución de roles musicales específicos que se definan en torno a la práctica, como pasa en el jazz, el hardbop, el free jazz, el blues, por mencionar algunas. En ellas, se hace notar una clara distribución de roles en cuanto quién lleva la base rítmica, armónica o melódica. Esto necesariamente implica una estratificación marcada de los roles donde generalmente la tímbrica de los instrumentos y sus capacidades en registro, agilidad y destreza del mismo músico, marcan la pauta de quién lleva, quién sigue y quién acompaña generalmente una línea melódica a la cual están supeditados los diferentes elementos musicales. Si bien es cierto que, en prácticas como el jazz la construcción melódica puede estar supeditada por las líneas armónicas que desarrolla el pianista o el guitarrista, la tendencia es que sí hay una estratificación de roles. Incluso, en muchas ocasiones, es el nombre del solista (quien a veces es el compositor de la música) seguido por el número de acompañantes lo que denomina a la agrupación de varios músicos; esto, necesariamente supone una verticalidad en los modos de hacer música en dichas prácticas.

Todos estos roles, me hacen pensar al improvisador como alguien mutable, en constante cambio y transformación, alguien que se relaciona incluso con el concepto de *espaciotienpomateria* de Karen Barad quien lo ejemplifica desde el análisis de la especie pfiesteria piscicida ("fish killer") de la cual, después de dos décadas de in-

vestigación no se saben sus características más básicas, lo cual deja ver un hueco en el estado actual de las aproximaciones científicas para su estudio. Pfiesteria tiene la capacidad de transformarse tanto en planta como en animal distinguiendo así un aspecto mutativo no lineal que posibilita su participación "en relaciones causales heterogéneas temporales y no deterministas"(Barad, 2007) recreándose y ampliando sus fronteras, manifestando una performatividad en continua transformación y surgimiento. De modo similar puede ser pensado el improvisador y la misma praxis la cual muta y "se construye con y para el público" que presencia este acto de creación musical en tiempo–real.Matthews (2012) Así la improvisación abre un abanico de posibilidades donde los roles, si es que los hay por acuerdo mutuo, pueden ir cambiando y compartirse de formas muy orgánicas, siempre partiendo de la escucha profunda y de la construcción colectiva que la agrupación quiera generar a un nivel más amplio. Así mismo hablar de una sola práctica sería limitar todas las posibilidades que la improvisación libre permite, por ello, más bien habría que hablar de prácticas en las cuales cada improvisador o grupo de improvisadores configuran la praxis improvisativa. El juntarse, compartir, indagar, pensar y resonar en colectivo es el principio; de ahí, es posible pensar que la improvisación libre esta conformada por una comunidad interesada en generar prácticas de creación musical que giran en torno a la escucha profunda, la toma de decisiones en colectivo y la compartición de saberes y experiencias.

2.2. La forma en la música libremente improvisada

Debido a que la forma (entendido en términos compositivos) no es un elemento en el cual se profundice al hablar de improvisación libre, considero importante problematizar sobre este concepto. En este apartado analizaré dónde la ubican algunos improvisadores, así como las derivas y posibilidades que permite pensar a la improvisación desde esta perspectiva. Otras preguntas interesantes que emergen a este respecto son: ¿cómo hablar de forma en la improvisación libre?, ¿acaso existe el interés por crear forma en la improvisación libre o todo lo contrario?

> Como dice Evan Parker: "La improvisación crea su propia forma"; y de manera similar, Carl T.Whitmer: "En la expansión se genera la forma".

> Frank Perry, el percusionista: "Para mí, la improvisación ha significado la liberación de la forma para que pueda adaptarse más fácilmente a mi imaginación". (Bailey, 1980)

La improvisación libre prioriza el acto de la creación musical al momento sobre conceptos organizacionales e incluso, puede prescindir de un proceso dialógico para generar algún tipo de guía y acuerdos previos en torno al desarrollo de la misma. En este escenario, ¿es posible encontrar formas musicales similares en esta expresión musical a las presentes en las prácticas musicales de la tradición musical occidental contemporánea, las cuales en muchas ocasiones toman referentes de las matemáticas, la ciencia, la poesía o la literatura como elementos constitutivos?[9]

Algunos improvisadores consideran que la forma en la improvisación se encuentra en el proceso de hacer música en tiempo-real y al ser impermanente, su forma queda en la memoria. No hay transcripción, ni interés (generalmente) por fijarla para tocarse después, su naturaleza no tiene directrices previas, es transitoria, cambiante e inestable. Así, su forma permeable, abierta a la contingencia, se queda en el espacio de encuentro y creación colectiva, donde múltiples agentes afectan y son afectados por la misma. Al respecto Derek Bailey comenta:

> Tal vez he dado la impresión de que no hay plan anticipado, ni estructura general, ni "forma". La crítica contraría a la improvisación libre, casi el único tipo disponible, casi siempre apunta a los mismos dos o tres objetivos y el claro favorito de estos es la"falta de forma". Como el criterio para evaluar una pieza musical, cualquier pieza musical, es usualmente heredado de las actitudes y prejuicios transmitidos por los estándares de la rígida música Europea [...]. En ninguna parte se aferra con tanta terrorífica tenacidad al concepto de forma como un conjunto ideal de proporciones que trasciende el estilo y el lenguaje como por los defensores de la composición musical. "La necesidad de diseño y equilibrio en ningún lado es más imperativa que en la música, donde todo es tan fugaz e impalpable:

[9]Es decir, la utilización de fundamentos matemáticos aplicados a la composición musical o la creación sonora tales como la teoría de conjuntos, teoría de grupos, la probabilidad, la estadística, sistemas dinámicos, etc.

> – meras vibraciones de la membrana timpánica." [...] Incluso en aquellas partes de la composición contemporánea donde los tipos anteriores de organización general ya no sirven, se ejerce una gran cantidad de ingenio para encontrar algo en lo que la música pueda "basarse". Mitos, poemas, declaraciones políticas, rituales antiguos, pinturas, sistemas matemáticos; parece que cualquier patrón general debe ser impuesto para salvar a la música de su endémica informe. (Bailey, 1980)

En este proceso impermanente, que va y viene en torno a estados continuos de atención/dispersión, surgimiento/desaparición, estabilidad/inestabilidad desplegadas a través del sonido, una vez concluida una improvisación puede ser muy difícil, que no imposible, asir una forma clara en la memoria que explicite el fenómeno en términos estructurales. Habrá quien desde una atención plena a la improvisación o a través de una perspectiva de escucha fenoménica, pueda abstraer la forma musical y no solo eso, sino destacar otros aspectos que fueron relevantes en esa improvisación. Sin embargo, en todo momento, la forma sigue siendo múltiple y espontánea.

Pese a la intrincada relación de características sonoras desenvueltas en el tiempo, la idea de coherencia, propósito, sujeción y dirección, siguen siendo importantes. Sin embargo, aunque la improvisación libre no es la única, ciertas aproximaciones a la creación musical no parten de contar una historia o intentar generar una narrativa coherente que lleve al escucha por un camino con un punto de llegada, sino que el foco de atención está en la materia sonora per se, esa masa de densidades sonoras y cambios en el tiempo que permiten a la consciencia entrar en estados donde podemos contemplar una forma móvil y desplegable mientras está sucediendo. Sin estos elementos, ¿Cuáles son los elementos sonoros y contextuales de la improvisación libre que dirigen la percepción de una performance como ésta? y, por otro lado, ¿en dónde ponen la atención tanto improvisadores como escuchas al momento de crear y presenciar esta música compleja? Para atender a esta música, considero necesario activar una atención multinivel que a través de una serie de experiencias sonoras, musicales y contextuales individuales, puede decantar en uno u otros lugares de atención hacia el momento sonoro presente. Ya sea que la atención se coloque en las amalgamas tímbricas, producidas por uno o más improvisadores a corto, mediano o largo plazo; en las

oleadas de la densidad textural, es decir, en los valles y crestas energéticos en términos de amplitud, brillo y densidad espectral; en los esquemas de interacción entre improvisadores, las coincidencias, desconciertos, puntos de inflexión, diálogos, aciertos y/o rupturas, y, de todo ello, los elementos que prevalecen y/o desaparecen: la novedad, la monotonía, la combinación, la transmutación; la atención hacia el paisaje sonoro circundante, el cual de manera consciente o inconsciente, tiene una repercusión que necesariamente modifica el espacio acústico contextual de la improvisación.

Si bien, con un alto grado de atención podríamos activar desde nuestra escucha una atención multinivel a través de una escucha entrelazada de dos o más objetos sonoros, resulta muy retador mantener este tipo de atención a todos y cada uno de los detalles que conforman una atención holística del fenómeno sonoro; más aún, cuando la velocidad gestual es un factor constructivo determinante. Esto puede complicarse un poco más en la presencia de elementos electrónicos y acústicos sucediendo paralelamente.[10] De este modo, la suma de todos estos elementos entrelazados generan un fenómeno emergente que es más que la suma de sus partes individuales. Con esto no quiero caer en conclusiones universalizantes que aplanen y coloquen todo a un mismo nivel y color. Evidentemente hay una gran variedad y diversidad de perspectivas y de formas de atender la realidad. Hay que buscar en todo caso una ontología muy amplia de donde partir, una donde quepa la infinidad interseccional de posibilidades de cada individuo. Al respecto me cierro esta idea al parafrasear y citar algunas ideas de Jacques Ranciere: siempre va haber un resto, alguien que se quede fuera, ese "sujeto político" que encarna "la parte de los sin parte", que no se identifica con un grupo social. Así que no podemos "universalizar las capacidades de cualquiera".(Rancière y Vila, 2012)

Si bien es cierto que cada persona atendería de forma diferente los elementos de una improvisación libre algunos de los elementos que detecto dentro de mi práctica como improvisador y como escucha son la atención desde una perspectiva más amplia hacia la forma en que la improvisación se construye, pasando por las frases, los periodos, los gestos largos, medianos, cortos y cortísimos, la densidad sonora,

[10]Un ejemplo de esta mezcla instrumental sería el noneto electroacústico Evan Parker `https://youtu.be/cjnmlyiewCI`

las oscilaciones cíclicas de la energía individual/grupal, las cualidades tímbricas y la espectromorfología de sonoridades individuales y amalgamas tímbricas, la novedad/-monotonía, los aciertos/desconciertos en la interacción entre los improvisadores, las maneras de interaccionar/responderse unos a otros, la forma de escucha que se presta al paisaje sonoro circundante y al propio proceso de conformación de la improvisación, etc. Todos estos elementos en interacción (y posiblemente muchos otros que no me he parado a reflexionar) son capaces de conformar mi atención hacia el fenómeno de la creación sonora solista o colectiva dentro de la práctica de la improvisación libre. Para que una máquina pueda llegar a generar una conformación holística de la escucha a este nivel y todo lo que involucra el proceso sutil de atención (bagaje contextual, emocional, personal, estético, atención, concentración, esfuerzo, etc.), se requeriría toda una vida de exploraciones o incluso más para llegar a generar una comprensión artificial que contemple el entrelazamiento de cada uno de los niveles de atención mencionados. Por ello, de la manera más honesta propongo no concentrar la atención en cómo podrían interrelacionarse todos estos niveles de atención de forma automática, sino solamente tomar los elementos ligados al desenvolvimiento acústico, gestual y formal de algunas músicas libremente improvisadas. Al intentar dirigir desde esta perspectiva entrelazada la atención hacia esta expresión musical, no estoy diciendo que otras músicas queden exentas del proceso complejo de atención que exige de por sí el hecho de escuchar, ni que todas las improvisaciones libres requieran de un proceso tan complejo de atención para poder discernir elementos clave que nos den pistas sobre su contenido y proyección formal. Incluso, la propia escucha del paisaje sonoro ya implica un proceso de atención que, dependiendo de las herramientas con que contemos, podemos escuchar desde este esquema de atención multinivel.

Desde esta mirada, qué queda en la memoria cuando las improvisaciones parten de un continuo fluir de secciones entrelazadas por elementos sonoros que aparecen y desaparecen en constructos orgánicos coherentes, donde la impermanencia del evento se convierte en una constante y es un elemento constructivo crítico de este método de creación musical. La improvisación es una música que se construye al momento *con y para el público* menciona el improvisador Wade Matthews (Matthews, 2012). En este sentido, si nuestra memoria y capacidades de escucha multinivel no están lo suficien-

temente desarrolladas, desde dónde es posible apreciar esta música. Si bien no es un requerimiento estar superdotado en términos de atención, escucha y memoria, para poder disfrutar y apreciar esta práctica musical, una posible respuesta sería partir de la contemplación de la impermanencia. Dejarnos maravillar por esos fenómenos contingentes que se convierten en otros, tomando a cada instante el lugar del elemento constructivo principal de esta música, nos exige entrar plenamente en el momento presente y atender de cada detalle sonoro/gestual que emerge de la sensibilidad hacia ese proceso de atención profunda. Pensar en que todo es impermanente puede servir de ayuda para colocarnos en un estado de atención plena desde el cual, es posible apreciar el valor del tiempo presente; ya sea desde su fragilidad, inconsistencia o desde la percepción de su permeabilidad desplegada en el tiempo como una estela sonora en constante surgimiento y dispersión. Mientras la memoria persista, estas estelas sonoras que dibujan de manera entrelazada el sonido, el pasado puede coexistir con el presente y posiblemente tender puentes que lleven la imaginación hacia rumbos latentes de la improvisación.

Es así que la estructura la pienso como una parte indivisible del sonido que moldea, mueve la escucha y genera diversos estados perceptivos a través del diálogo en continua transitoriedad con todos los elementos que la conforman.

> La idea de que todos estos fragmentos existen por separado es, evidentemente, una ilusión, y esta ilusión no puede hacer otra cosa que llevarnos a un conflicto y una confusión sin fin. Es más, el intento de vivir de acuerdo con la idea de que estos fragmentos están realmente separados es, en esencia, lo que nos ha llevado a la creciente serie de crisis sumamente urgentes que hoy se nos están planteando. (Bohm y Apfelbaume, 1998)

Como señala Bohm, no hay separación ni fragmentación, aproximarse así a la experiencia de la realidad es una construcción respaldada bajo el paradigma racional de la física clásica. Del mismo modo, posicionar la atención solamente en la forma, deja de lado la conjugación inmensa que podrían generar, al posicionarla de manera holística, cada uno de los elementos sonoros que componen una improvisación o un momento sonoro. Desde ahí, es posible transitar por un continuo de masas y

densidades sonoras entrelazadas a distintos niveles, en un flujo y transformación ininterrumpida, ya sea desde la perspectiva de una obra (abierta o cerrada pero escrita) o una improvisación. Siguiendo los planteamientos de (Bohm y Apfelbaume, 1998), señala que:

> [...] el modo general en que el hombre concibe la totalidad, es decir, su visión general del mundo, es crucial para el orden total de la propia mente humana. Si concibe la totalidad compuesta por fragmentos independientes, así es como su mente tenderá a funcionar, pero si puede incluirlo todo de una manera coherente y armoniosa en una totalidad general que sea indivisa, ininterrumpida y sin fronteras (porque cada frontera es una división o fisura), su mente tenderá a moverse de una manera semejante y de ahí fluirá una acción ordenada dentro del todo. (Bohm y Apfelbaume, 1998)

La forma contiene aparentemente al caos de un sistema. Algunos sistemas matemáticos caóticos pueden llegar a ser deterministas al conocer perfectamente sus condiciones iniciales, así el caos en términos matemáticos puede contenerse y presentar formas claras, complejas pero finalmente determinadas. Ello puede ser observable en el atractor de Lorenz (efecto mariposa), un sistema de ecuaciones diferenciales ordinarias que él mismo propuso para analizar las ecuaciones dinámicas de la atmósfera terrestre. Debido a la multiplicidad inherente que presenta el caos, es a través de la forma que se vuelve reconocible y asequible a la percepción; así el caos resulta en la forma.

Sin embargo, si dichas condiciones iniciales, incluso en un sistema matemático caótico, varían tan solo un poco, el sistema puede presentar comportamientos completamente diferentes, llevándolo a estados intrínsecamente inestables, donde la multiplicidad de factores implicados, vuelve imposible su predicción. El problema es que las variables internas y de entrada, no pueden conocerse a la perfección para poder determinar claramente las salidas de un sistema, sus estados futuros y menos aún su forma. El físico David Deutsch escribe:

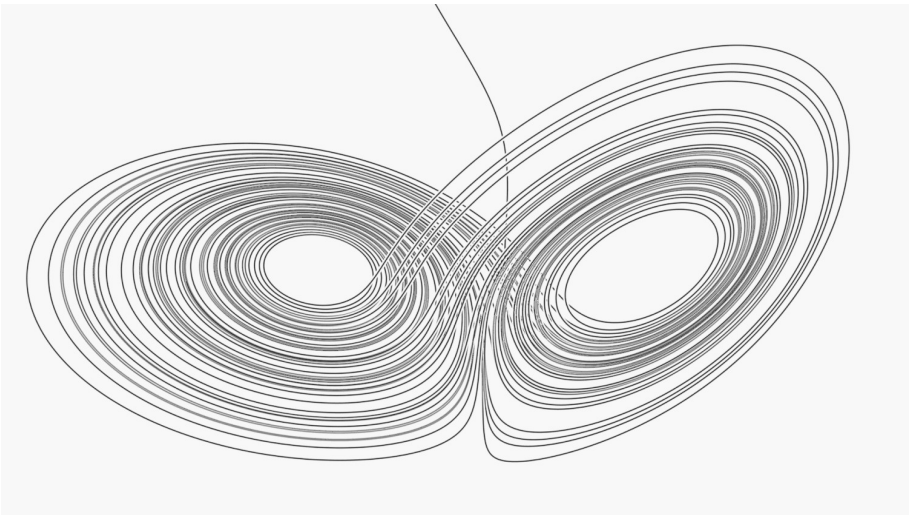

Figura 2.1: Atractor de Lorenz.

> ¿Conociendo estas condiciones iniciales con un grado aceptable de exactitud, podríamos predecir con igual aproximación el comportamiento futuro del sistema? La respuesta es que, por lo general, no. La diferencia entre la trayectoria predicha, que ha sido calculada a partir de datos ligeramente inexactos, y la real tiende a crecer de modo exponencial e irregular («caóticamente») con el tiempo, de forma que el estado inicial, imperfectamente conocido, acaba por no servir de guía para el comportamiento del sistema. Ello implica, por lo que se refiere a las predicciones realizadas mediante ordenador, que los movimientos de los planetas, epítome de la predecibilidad clásica, son sistemas clásicos atípicos. [...] Nuestra incapacidad para medir de manera adecuada las condiciones iniciales resulta por completo irrelevante. Aun en el caso de conocerlas a la perfección, se mantendría la multiplicidad y, por consiguiente, la impredecibilidad de la evolución. (Deutsch y Sempau, 1999)

Desde la perspectiva conceptual y racional con la que partimos para percibir y nombrar el mundo, se atañen desde su consolidación ideas, unificaciones, reducciones, sesgos y atisbos que cohesionan nuestra experiencia caótica sesgando la forma de concebir la realidad, limitada por una serie de cánones y memorias simbólicas inherentemente heredadas. Aún así, la estructura es la que observa y es observada, la que da vida a las intenciones y la maleabilidad que prevalece en el mundo, no solo desde la

Figura 2.2: Proporción áurea en la naturaleza, fenómenos naturales y creaciones humanas.

integración de las partes sino como la suma de todas ellas para generar un ente completamente diferente a la forma y el contenido de ésta.[11] En constante intra-acción, todos los elementos se amalgaman para conformar en el espacio-tiempo la estructura caótica y azarosa de la realidad y sus derivaciones en la transdimensionalidad del multiverso.[12]

Al igual que la multiplicidad inherente presente en los sistemas naturales y sociales, en la improvisación libre es posible hallar una multiplicidad de factores implicados en este sistema artístico, entre ellos, las formas móviles que van y vienen, que son caprichosas, abiertas a la contingencia y sujetas a la afectación que propone la conjugación de los improvisadores, el entorno acústico y la presciencia de un público. Tomando estos factores en cuenta: ¿en qué se convierten las improvisaciones libres una vez que son encapsuladas por cualquier medio de grabación, y reproducidas,

[11]"no se puede mantener la división entre el observador y lo observado (que está implícito en la visión atomista que considera a cada uno de estos como agregados separados de átomos). Más bien, tanto el observador como lo observado son aspectos fusionados e interpenetrados de una realidad total, que es indivisible e inanalizable."(Bohm y Apfelbaume, 1998)

[12]"[...] la mayoría de los físicos prefieren seguir utilizando la palabra «universo» para denominar a la entidad de siempre, aun cuando ésta resulte ser ahora sólo una pequeña porción de la realidad física. Un nuevo término, multiverso, ha sido acuñado para denominar la totalidad de la realidad física."(Deutsch y Sempau, 1999)

cristalizándose en un soporte fijo? ¿Representa esto un problema para la propia esencia transitoria de la improvisación? ¿Qué supone la capacidad técnica de nuestros dispositivos para reproducir infinitas veces un acto que en principio surgió de la contingencia y la interacción con una serie de factores que ciertamente afectan el proceso de creación de una improvisación? Más sencillo, ¿qué es y qué posibilita la improvisación libre una vez que ha sido grabada?, ¿deja de ser una improvisación libre, que se construye *con y para el público*, para convertirse en un producto de la reproducibilidad técnica? ¿Será que la improvisación una vez grabada pierde esa cualidad de inmediatez, ese aura relacionado con la experiencia de lo irrepetible para convertirse en un producto?

Si bien el improvisador, compositor e investigador Josep Lluís Galiana tomando aseveraciones de Wade Matthews quien a su vez recurre a una cita del músico de jazz Eric Dolphy: *"Music, after it's over, it's gone in the air. You can never catch it again"* señala que:

> La improvisación libre es un proceso creativo y no un producto acabado, manufacturado, listo para ser escuchado y consumido tantas veces como se desee. Al mismo tiempo, la ausencia de producto en la improvisación libre conduce indefectiblemente a pensar en la ausencia de un compositor, que entiende la creación musical como la elaboración de un producto en dos fases: la escritura de una partitura y la ejecución de la misma por un intérprete. En la improvisación libre, en cambio, se invierte la relación entre el proceso de creación y lo creado, que es el producto. Por lo que el musicólogo no puede estudiar el producto, es decir, la partitura, su contextualización histórica y sus diversas connotaciones estéticas y referencias estilísticas, puesto que no existe nada antes del momento mismo de poner en marcha el proceso creativo. Creadores y público no cuentan con un objeto al que acudir, todos ellos han de compartir el proceso creativo en el mismo momento y lugar, y cuando se dé por finalizado todo se habrá desvanecido. Cabe concluir, por tanto, que el proceso es el único producto posible en la improvisación libre. (Galiana Gallach, 2018)

Dadas estas afirmaciones, cabe mencionar que, lo que podríamos encontrar como producto acabado una vez finalizada una improvisación sería su registro por cualquier medio digital o analógico, convirtiendo la improvisación en un objeto que es posible conservar *a posteriori*. Su grabación sería ese producto que existe y se cristaliza después del proceso de la improvisación para su futura escucha o indagación musicológica. A este respecto Wade Matthews menciona:

> "[...], si Parker viera en la grabación de una improvisación su plasmación como obra fija, estaría diciendo que cuando se graba es una composición, y cuando no, pues no lo es. Eso es pintar con una brocha demasiado gorda, porque el hecho de que una grabación sea algo fijo no convierte la improvisación en algo fijo. Una grabación de una improvisación tiene la misma relación con esa improvisación que una fotografía con la persona fotografiada. Se trata de la relación entre un documento fijo e inmutable y algo vivo y en constante cambio."(Matthews, 2012)

Definitivamente una vez que se hizo un registro aural de una improvisación, esta adquiere la cualidad de producto, un objeto estético cosificable en su dimensión política por espacios físicos y digitales de mercado. Incluso, en algunos casos, las grabaciones de improvisaciones libres, cambian su estatus para adquirir el de composición u obra; permanente, reproducible y acabada. De este modo es posible establecer certezas porque finalmente lo impermanente se pudo cristalizar, meterse en repositorios y etiquetar toda la complejidad aparente del carácter de la improvisación. En ese sentido, la improvisadora e investigadora Chefa Alonso se ha referido a la improvisación como *la composición en movimiento*, aludiendo a que ésta, en algunos casos, puede partir de una serie de elementos previamente acordados, en mayor o menor medida, entre los músicos que participan en una improvisación. Un ejemplo de cómo la improvisación se convierte en composición una vez que se fija en algún medio de reproducción es el álbum *Open Paper Tree* de Michel Doneda, Paul Rogers, Lê Quan Ninh.[13] Aunque los tres músicos se consideran improvisadores, el disco aparece en algunas plataformas digitales con el apelativo de "música compuesta por:" en el entendido de que los formatos de dichas plataformas y mecanismos de derecho de autor,

[13] `https://destination-out.bandcamp.com/album/open-paper-tree`

no contemplan que la música puede ser catalogada como "música improvisada por:". Otro ejemplo en el que las improvisaciones (diálogos) adquieren el carácter de obra son los *DIÁLOGOS - creación colaborativa - serie 4*[14].

A este respecto, Wade Matthews comenta, "[...] la música [improvisada] se está creando en el mismo instante en que se escucha. No es una recreación, la interpretación de una partitura compuesta hace una semana, un mes, o incluso ochenta años".(Matthews, 2012) En este sentido, de qué forma sería pertinente acercarnos al registro y encapsulamiento en el tiempo de un fenómeno como la improvisación libre.

Afirmar que hay alguna alternativa en la que se podría devolver el aura de la impermanencia a la libre improvisación una vez que ha sido grabada, sería romantizarla y limitar sus posibilidades a algo único e irrepetible que solo pasa in situ. De ahí, considero importante plantearse cómo desmitificar esta idea y más que devolverle su aura de inmediatez e impermanencia que caracteriza la creación de esta expresión musical, cómo aproximarse a ella desde su registro. Aunque esto podría aplicar a otras músicas no necesariamente improvisadas, una forma de dialogar con la impermanencia registrada y reproducible, es prestar una escucha profunda a cada instante de esa música que fue creada en interacción con un espacio y contextos que ciertamente ya no están en el momento de su posterior reproducción. Para una escucha atenta, cada momento de la reproducción podría convertirse en un momento único e irrepetible, caracterizado por todo el contexto y experiencias tanto previas como presentes que se experimentan al momento de escuchar una improvisación grabada. Nada permanece igual, nunca vamos a presenciar lo que aparentemente es lo mismo de la misma forma o, en palabras de Heráclito de Éfeso, «Nadie puede bañarse dos veces en el mismo río».

Dicho todo lo anterior, la improvisación no carece de algún tipo de gradación estilística. Si bien, a detalle, cada improvisación es única e irrepetible, y aquí también podríamos incluir estos adjetivos a cualquier interpretación musical ejecutada por un humano, es un hecho que hay rasgos y criterios estilísticos que delimitan certeramente lo que se puede hacer y lo que se espera estéticamente en una improvisación. No obs-

[14] `https://bit.ly/3K1rmzU`

tante, hay un espectro muy amplio de posibilidades en las prácticas de improvisación que llenan de matices y contrastes a esta práctica.

Partiendo de esta perspectiva surgen varias preguntas: ¿habría una diferencia entre el escuchar/analizar música académica contemporánea[15] y la improvisación libre? ¿Sería posible distinguir en términos numéricos (a través del análisis de la señal de audio) qué es y qué no es una improvisación y de ahí, crear un descriptor de audio numérico que nos permita medir el índice de *improvisatoriedad* o *improvisabilidad* en distintos tipos de música? ¿Será que a partir de su análisis pueda arrojar algunas perspectivas que engloben a la libre improvisación en un espacio delimitado por ciertos criterios, diferentes a otras prácticas musicales?

Para responder a estas preguntas, y hablar de la forma en la improvisación libre, considero necesario colocar al centro de este problema una perspectiva que parta de analizar los diferentes parámetros sonoros e interactivos que permitan ahondar en éste término. Por un lado, desde las relaciones sonoras implicadas; como las estructuras generadas en términos de novedad/monotonía desplegadas en la amplitud, el brillo, la complejidad, la densidad y la centralidad espectral. Por otro, de la evolución a nivel energético y de las intra-acciones suscitadas entre cada uno de los agentes que conforman el "sistema improvisación libre". Estos elementos los tomo como punto de partida para rastrear relaciones a distintos niveles temporales (a largo, mediano y corto plazo) en la música libremente improvisada. Estos elementos serán abordados en el siguiente capítulo para dar cuenta de cómo a través de las descripciones numéricas del audio y su análisis en forma de series de tiempo, mediante algoritmos que permiten hacer predicciones a futuro (Long-Short Term Memory), es posible rastrear esas relaciones temporales que pueden remitir directamente no solo al análisis sino a la recreación en tiempo-real de la forma en la improvisación libre. La idea central es que a través de esta perspectiva, sea posible entrenar un modelo computacional para reconocer ciertos aspectos musicales en una práctica tan impredecible como es la improvisación libre, y además se puedan obtener resultados predictivos en términos formales que ayuden a descifrar algunas particularidades de la improvisación libre. Para responder a otras preguntas derivadas en esta sección, en el siguiente capítulo

[15] `https://es.wikipedia.org/wiki/Música_clásica_contemporánea`

abordaré a través del análisis de algunas improvisaciones tanto mías como de algunos improvisadores, los problemas, soluciones, derivas y aciertos que he explorado en torno al tema de la predicción de la forma en la improvisación libre.

2.2.1. Entrelazamientos en la escucha profunda de humanos y máquinas

La idea de profundidad desde la perspectiva de la compositora Pauline Oliveros, tiene que ver con lo complejo, con la forma en que se empujan ciertos límites de la percepción para llevarla a un espacio de frontera donde es posible discernir espacios acústicos que normalmente no estaban y que van más allá del entendimiento habitual. Aquí el espacio acústico es entendido como el lugar donde el tiempo y el espacio se mezclan siendo articulados por el sonido. Partir de la escucha profunda que propone Pauline Oliveros me es de suma utilidad para abordar una práctica artística como la improvisación libre desde una perspectiva fenomenológica como es la escucha profunda.

La profundidad que hace Oliveros también se liga con la idea de alguien o algo profundo que es difícil de comprender debido a que se requiere una gran concentración/atención para acceder a sus contenido. Para Oliveros escuchar profundamente significa "aprender a expandir la percepción de los sonidos para integrar todo el continuo del espacio/tiempo sonoro, encontrando la vastedad y la complejidad tanto como sea posible."(Oliveros, 2005) La escucha profunda involucra la focalización sonora hacia un sonido o una serie de sonidos materializados en el continuo del espacio/tiempo y permite percibir el detalle o la trayectoria de ese fenómeno sonoro o fenómenos sonoros. Además ese enfoque de la atención debe regresar a la forma de percepción más amplia del contexto sonoro, donde esa expansión perceptiva nos ayude a conectar con todo el entorno e incluso más allá de este. En palabras de Oliveros: "El nivel de conciencia del paisaje sonoro provocado por *Deep Listening* puede conducir a la posibilidad de dar forma al sonido [/escucha] de la tecnología y de los entornos urbanos."(Oliveros, 2005) La escucha profunda posibilita la idea de escuchar al mundo como una gran composición sonora que se va modelando a partir del enfoque aplicado a la sucesión sonora, que permanece en continua formación. La escucha profunda es

también una forma de meditación, donde la atención es dirigida hacia la dinámica de sonidos y silencios coexistiendo en el espacio acústico, el sonido es entendido más allá de su contexto musical o vocal que incluye esa parte vibracional de formaciones sonoras complejas y en constante relación.

Esta perspectiva de escucha es de suma utilidad para entablar un entrelazamiento con el espacio acústico partiendo de lo que ya existe a nivel sonoro para entender cómo podemos vincularnos a éste, transitando siempre de la escucha hacia la producción sonora. En las prácticas musicales, específicamente en la performance, la improvisación, el análisis y la composición, considero fundamental partir de la escucha profunda para, en palabras de Oliveros "expandir la conciencia de lo sonoro en tantas dimensiones de conciencia y atención dinámica como sea humanamente posible."(Oliveros, 2005)

Por otro lado, si hay una escucha profunda en humanos, ¿sería posible hablar de escucha profunda de máquinas o automática? La respuesta corta es un no rotundo por más "Deep leaning" que sea el sistema en cuestión o por más gigantesca que sea su base de datos. En el caso particular de SEALI, al momento de conformar una base de datos para comenzar su proceso de análisis, hay un proceso de "transferencia de conocimiento" que hago partiendo de una escucha profunda de las grabaciones de improvisaciones libres a utilizar. En ellas podríamos decir que se encuentra el bagaje de habilidades técnicas ligadas a la escucha y la ejecución de los improvisadores. Desde aquí ya se podría hablar de un sesgo tecnológico implicado en el proceso, ya que la grabación involucra el procesamiento del fenómeno acústico a una señal digital. Además, posteriormente el proceso involucra la segmentación de los ejemplos en unidades gestuales discernibles a nivel tímbrico (esta por demás decir que aquí hay otro sesgo o de reducción en la información sonora sumamente importante). Después, hay una extracción multidimensional de características del audio las cuales categoriza para realizar una distribución interna que le permite establecer sucesiones numéricas –en forma de sonido– a lo largo del tiempo. Ello haciendo uso de algoritmos de redes neuronales profundas y memoria a corto-largo plazo (LSTM). Todo este proceso de digitalización de la escucha, necesariamente incluye múltiples sesgos, tanto técnicos como socioculturales en cada proceso, que van quitando capas a la complejidad im-

plicada en el proceso de escucha humana. Además, es cierto que muchas de estas herramientas han surgido debido a los avances en las tecnologías de grabación, reproducción y visualización sonora que el estudio de la la fisiología humana, la acústica y la psicoacústica han aportado, pero ello no significa que sea posible hablar de similitudes en procesos que ciertamente, de fondo, no sabemos cómo operan a nivel interpretativo en el cerebro. También es cierto que la escucha profunda se inserta en los avances tecnológicos ligados a la grabación de campo, la reproducción y la escucha digital, desde esta posibilidad podemos acceder a un nivel microscópico de escucha, "Grabar el flujo de sonidos a través del continuo espacio/tiempo como un periodista puede promover una comprensión más profunda de su presencia y significado en el entorno"(Oliveros, 2005) sin embargo, esta forma de percepción humana tampoco tiene que ver con cómo la máquina opera con la información del audio y establece relaciones numéricas.

Sin embargo, a diferencia de nosotros, el momento de entrenamiento de una máquina con una base de datos involucra varios procesos de cálculo automatizado, análisis profundo y filtración de información, este estado de atención y análisis lo podría mantener por un lapso de tiempo indefinido, permitiendo ejecutar esta acción sobre grandes bases de datos en tiempos infinitamente menores a los que tardaríamos en escuchar y analizar la misma cantidad de información esto significa que las máquinas podrían escuchar, no de forma profunda, pero sí mucho más de lo que como humanos podríamos escuchar en toda una vida.(Collins, 2016) En este sentido resulta interesante abordar las posibilidades que las máquinas nos pueden aportar al análisis de grandes caudales de información. Encontrar un punto medio entre el uso de éstas herramientas, las posibilidades intrínsecas a su uso y lo que podemos aportar para la conformación de una escucha automática más fiel, precisa, ligada a lo que atendemos en una improvisación, resultaría crucial para indagar sobre las posibilidades expansivas y la exploración de nuevos hallazgos en la conformación de un espacio de cooperación humano-máquina. Desde ahí considero que sería más preciso hablar de una escucha profunda humana expandida, hibridada por las posibilidades que las máquinas aportan a los procesos de atención humana y en el caso de la máquina como un artefacto capaz de escuchar bajo sus propios términos y niveles de análisis,

los cuales para muchos casos pueden ser útiles y para otros no. Sin embargo el fenómeno de la afectación y la intra-acción permanece vigente, lo cual mantiene abiertas las posibilidades de transformación y entrelazamiento de agentes humanos y más que humanos.

Capítulo 3

SEALI: Sistema de Escucha Automática para la Libre Improvisación

En este capítulo abordo las funcionalidades de SEALI el cual es un sistema sonoro interactivo que parte de la escucha artificial, modelada por mi propia escucha, para analizar elementos sonoros en un ámbito muy acotado de la improvisación libre e interactuar a varios niveles temporales mediante la clasificación y la predicción sonora con otros improvisadores en el contexto de dicha práctica. Estás funcionalidades las pienso como algoritmos que en conjunto articulan la arquitectura de SEALI.[1] Las funcionalidades involucradas, requieren de técnicas de selección, sistematización, estructuración y análisis del audio para poder generar una serie de supuestos abstractos a nivel numérico que describan esta práctica musical. Cada una de ellas implica varios procesos entrelazados por bucles de retroalimentación donde mi escucha y el aprendizaje de máquina se unen con el objetivo de desarrollar la metodología y las técnicas algorítmicas que posibilitaron la realización de SEALI y del presente documento.

[1]Debido a la flexibilidad inherente de los algoritmos, a sus características abiertas al cambio y modificación continua, éstos podrían adecuarse a diversos contextos y emplearse en un amplio espectro de aplicaciones, ajustándose no solo a la improvisación libre sino a muchas prácticas; incluso no musicales.

Siguiendo con las funcionalidades de SEALI, lo primero que requiere es un corpus que le proveerá el "conocimiento" sobre la improvisación libre (o la práctica sonora/-musical que se quiera analizar). Una vez conformado este corpus, el siguiente paso es su análisis, para lo cual es necesario segmentarlo y construir una base de datos a través de funciones capaces de abstraer diferentes características de las señales digitales del audio de los segmentos presentados; a estas características también se les conoce como descripciones de audio (Vinet *et al.*, 2002). Posteriormente, se utiliza un algoritmo de aprendizaje automático para analizar la base de datos. El algoritmo permite diferenciar, a través de la actualización de sesgos (bias) y pesos (weights) las particularidades de la base de datos. A través de las épocas de aprendizaje indicadas al algoritmo, éste produce una función inferida que le permite llegar a generar la certeza necesaria para clasificar o predecir diferentes datos en relación a los que utilizó durante el proceso de entrenamiento. Para ello, la base de datos se divide en dos partes, una que es utilizada para entrenar y otra que es para probar el modelo computacional generado en el proceso de entrenamiento. Si bien hay varios tipos de aprendizaje automático, SEALI utiliza dos perspectivas: el aprendizaje supervisado y el aprendizaje sin supervisión. La diferencia principal entre estos tipos de aprendizaje de máquina es que el primero recibe una base de datos anotada (análogo a una experiencia de aprendizaje pasivo basada en el conocimiento de alguien más) y en el segundo, recibe los datos sin clasificar (similar a un aprendizaje experiencial). Para cualquiera de los dos modos de aprendizaje, es necesario construir una base de datos numérica con la cual los algoritmos de aprendizaje puedan entrenar sobre los elementos presentados, y, posteriormente, predecir elementos diferentes a los utilizados en el proceso de entrenamiento del algoritmo.(Fiebrink, 2011) En el caso del análisis de la forma, las bases de datos generadas son empleadas para aprender su estructura y predecir posibles estados futuros dada una secuencia inicial. Para ello, los datos son colocados en series de "n" pasos de tiempo *(timesteps)*, para ser analizadas por una red neuronal recurrente. Todo esto le permite a SEALI diferenciar las particularidades tímbricas y sistematizar las estructuras numéricas del corpus, a partir de allí, se crea un modelo computacional que puede ser ejecutado en tiempo–real para realizar nuevas clasificaciones o predicciones. Finalmente, los datos generados por el modelo son enviados a un segundo algoritmo que puede gestionar los datos y accionar cier-

tas respuestas sonoras/silentes congruentes con el contexto sonoro en un momento determinado durante una improvisación libre.

Desde una perspectiva más conceptual, parto del concepto de agencialidad algorítmica o algoritmicidad[2] para analizar cómo se inserta en los diferentes procesos dentro del ciclo de retroalimentación que contempla a la práctica artística, el desarrollo tecnológico y la investigación. Asimismo, analizo sus implicaciones en los objetivos de la presente investigación y los resultados sonoros que se generan derivados, aparentemente, de mis propias decisiones, ya que los sesgos algorítmicos permean todo el proceso, desde la micro hasta la macro-escala. Las funcionalidades de SEALI pasan por un proceso de afectación constante, incidiendo de manera activa en la forma en que segmenta y clasifica la información que recibe, las predicciones de nuevos datos y el resultado general que produce la intra-acción mutua de todas las partes implicadas en el proceso. Debido a la escucha artificial de SEALI, es decir, su forma de analizar, sistematizar, clasificar, predecir y entregar nuevos elementos en la improvisación, encuentro interesante pensarla como una forma de escucha trans-humana donde la metáfora de una máquina que escucha e improvisa, cobra sentido al posibilitar la transducción de mi propia forma de escuchar, hacer y concebir la música libremente improvisada. Esta transducción está condicionada por la retroalimentación humano-algorítmica que ha influenciado a la configuración del sistema a través de varios procesos recursivos. La escucha artificial de SEALI está sujeta por varios aspectos: su *conocimiento* (la base de datos con la que se entrena, seleccionada por mí mismo) y el sesgo algorítmico del que parte, el cual, muchas veces viene desde la programación inicial de los algoritmos. Además, a ello se suman los múltiples bucles de retroalimentación mediante procesos de validación e interacción humana. Derivado de ello, fue inevitable transitar por múltiples procesos de revisión, adecuación, modificación, reelaboración y reestructuración, de los objetivos iniciales dada la influencia continua de las intra-acciones humano-algorítmicas.

Otra característica importante para el desarrollo de SEALI surgió de la búsqueda por dotarlo de una serie de funciones que le permitieran sistematizar la estructura sonora de la improvisación libre, y con ello, aproximarse a la forma musical de la

[2]Definida en el capítulo anterior

improvisación libre, a través del análisis de series de tiempo y de la ejecución de predicciones a futuro. El objetivo fue que a través de la sistematización de estas funciones, SEALI pudiera participar en la práctica de la improvisación libre. Partiendo de este análisis estructural, es posible observar un tipo de "transferencia de conocimiento", en términos numéricos, de las bases de datos con las que el sistema fue entrenado. Dicho conocimiento sobre estas estructuras podría ser transferido hacia otros contextos, tiempos y espacios. Es importante resaltar que los sesgos implícitos en SEALI siguen presentes a este respecto. La transferencia de conocimiento solo se adecúa a través de la predicción numérica y las filtraciones a las que fueron sujetos los datos de entrada, así como el mismo proceso comparativo entrada-salida y no precisamente una transferencia del conocimiento del propio estilo y aproximaciones improvisatorias. Ciertamente, una máquina no podría ser capaz de abstraer y aún menos comprender lo que ocurre en la improvisación libre, en todo caso podría hablar de la sistematización de una serie de operaciones que le permiten descifrar patrones superficiales del contenido de esta práctica musical. Además, para aproximarme a los resultados obtenidos, se tuvieron que realizar muchos procesos de abstracción en la información que inevitablemente quitan capas de resolución a las predicciones que SEALI genera. Sin embargo, como explicaré más adelante en éste y el siguiente capítulo, al limitar la escucha de la máquina a ciertas descripciones numéricas del audio, es posible obtener buenos resultados en las clasificaciones y predicciones que genera del audio, así como la generación de réplicas numéricas estructurales idénticas a las versiones grabadas de algunas improvisaciones libres con las que trabajé durante esta investigación.

A continuación presento una lista y dos diagramas generales de los puntos que fue necesario abordar para la conformación de las funcionalidades que componen a SEALI. Los cuales se encuentran divididos en dos partes que incluyen las funcionalidades de SEALI para la clasificación tímbrica, así como para la predicción de la estructura en la improvisación libre.

- Proceso de una escucha atenta y profunda sobre un amplio espectro de improvisaciones libres de los 60s a la fecha.

- Generación (grabación) o acopio de materiales sonoros que son la base del conocimiento del sistema. Esto también puede ser enriquecido o sustituido por un corpus de improvisaciones libres.

- Extracción de características de señales digitales de audio mediante descripciones numéricas.

- Generación de bases de datos anotadas mediante descripciones numéricas del audio a través de la clasificación sin supervisión por medio del algoritmo *Mean Shift Clustering* - MSC.

- Generación de modelos computacionales de aprendizaje automático supervisado mediante redes neuronales profundas y redes neuronales recurrentes que abstraen la información del punto anterior para su clasificación individual y la predicción secuencial de datos mediante series de tiempo.

- Sistema que incluye la ejecución en tiempo–real de los modelos computacionales creados en el punto anterior para la identificación y la predicción estructural de nuevos materiales sonoros.

- Mecanismos interactivos entre SEALI y un improvisador o improvisadores libres, donde se incluye la generación de materiales sonoros, en la que se incluye la síntesis concatenativa.

A continuación presento cada una de estas secciones a detalle.

3.1. Corpus musical y base de datos

Antes de comenzar con las funcionalidades de SEALI, como he descrito anteriormente un elemento sumamente importante en este proceso fue partir de una atención y una escucha atenta a un amplio espectro de improvisaciones libres de los 60s hasta la actualidad. Esto me permitió no solamente escuchar los complejos sonidos producidos sino comenzar a entender esta práctica desde las múltiples modalidades que plantea. Esto fue de gran ayuda trazar un mapa mental más claro sobre los elementos

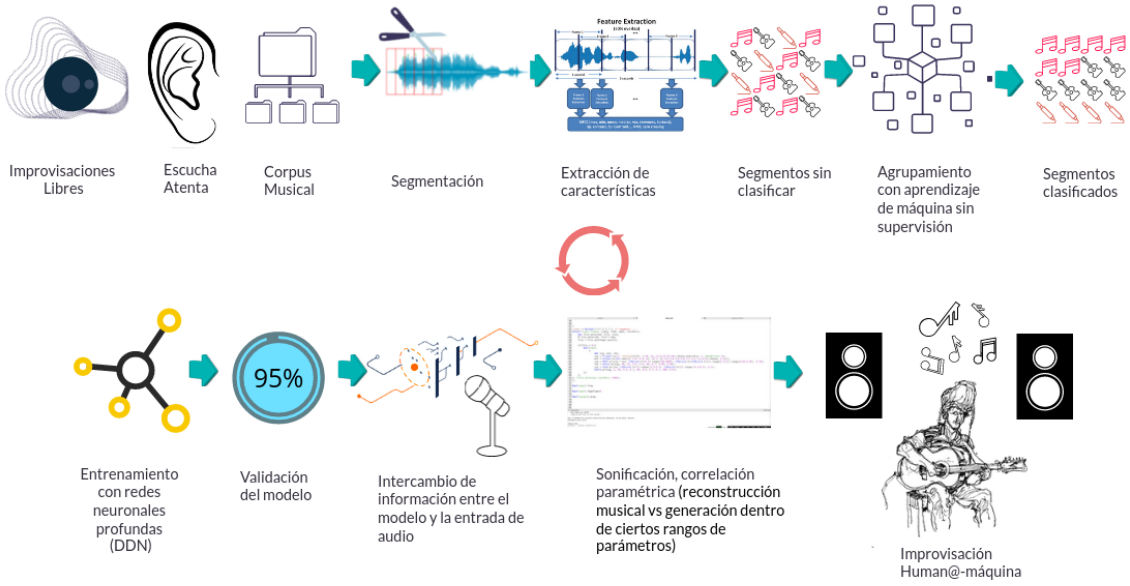

Figura 3.1: Diagrama de funcionalidades de SEALI para la predicción tímbrica de la improvisación libre.

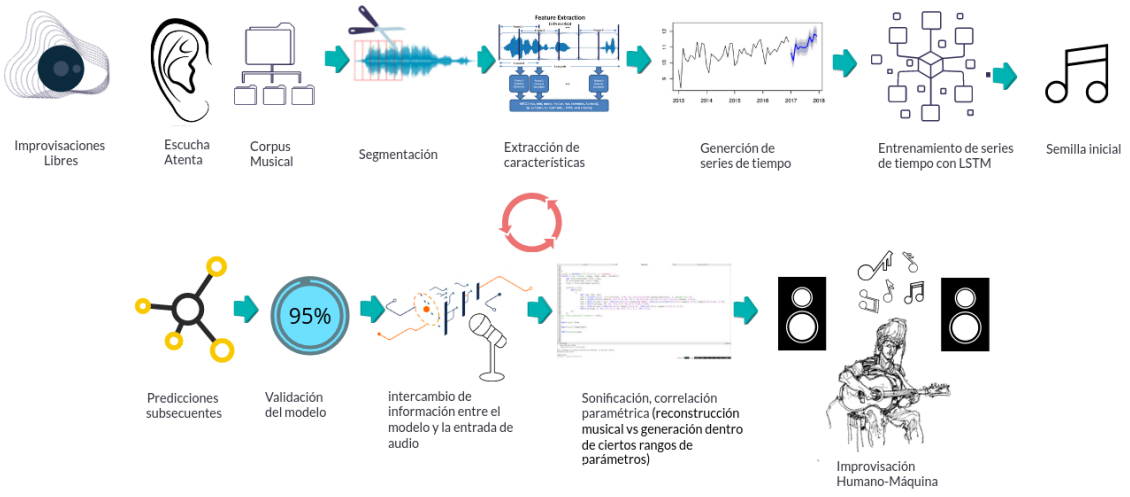

Figura 3.2: Diagrama de funcionalidades de SEALI para la predicción de la forma en la improvisación

que percibo de esta práctica y comenzar a delimitar las funciones que el sistema en cuestión requería.

Después de este proceso, las funcionalidades de SEALI siguen con la generación del corpus de improvisaciones libres que posibilitará dirigir sus clasificaciones y predicciones a lugares muy específicos. Para esto hay varias posibilidades. Se puede partir, de realizar grabaciones de ciertos gestos o aproximaciones técnicas de algún improvisador que sean significativas en su práctica. Esto con el fin de entrenar un modelo de aprendizaje automático que sea capaz de reconocer a nivel tímbrico las características más relevantes de su práctica. Aunado a esto, generar un corpus de improvisaciones libres preexistentes con el objetivo dotar a SEALI del conocimiento de las estructuras formales recurrentes en la improvisación libre. Estas dos posibilidades se pueden abordar de diferentes maneras y permiten explorar un amplio abanico de combinaciones que le pueden dar a SEALI ciertas características creativas para entrar en diálogo con otros improvisadores.

Como veremos en éste y el siguiente capítulo, el corpus musical fue cambiando a lo largo de la investigación como resultado de los procesos de afectación múltiple entre todos los aspectos implicados y da cuenta de cómo desde la experiencia de escucha de la improvisación libre se tienen que ajustar muchos de los elementos que componen el desarrollo algorítmico-investigativo. En este proceso se realizaron varios experimentos. El primer corpus con el que comencé, comprendió una gran variedad de grabaciones de diferentes improvisadores libres de Europa, Estados Unidos y México a partir de los 60's a la fecha; conformando un total de 350 archivos de audio que suman 29 GB y 45 horas de tiempo de reproducción.

En los siguientes apartados desarrollaré a profundidad cada uno de los aspectos que involucraron el procesamiento del audio para la conformación de una base de datos sin supervisión basada en el corpus seleccionado. Después, la misma base de datos, será empleada para convertirla en una base de datos anotada empleando los criterios de clasificación sin supervisión.

3.2. Criterios de Segmentación sonora

Para nosotros como humanos, así como para un sistema que "ve" o "escucha", es necesario abstraer las propiedades de un objeto en diferentes segmentos de información que ayuden a diferenciarlo; por ejemplo, si queremos identificar la imagen de una flor con visión computarizada, lo primero que se requiere es segmentar la imagen en píxeles y establecer ciertas relaciones entre todos los píxeles que componen la imagen. Además, el sistema necesita de muchos ejemplos en contextos diferentes, donde aparezca el objeto que queremos analizar para diferenciarlo de otros elementos que puedan aparecer en la escena. De forma semejante a la visión computacional, para aproximarnos a la escucha de máquina se requiere una aproximación que incluya la segmentación de todos los elementos que nos interesa analizar, en el caso de las prácticas musicales que abordo en esta tesis, un criterio para realizar dicha segmentación parte del análisis de gestos sonoros, entendidos como elementos individuales que por sí mismos, son cognoscibles auditivamente y contienen información que puede significar aunque se encuentre desligado del contexto que lo abarca. Los gestos podrían estar caracterizados por componentes como el timbre, la altura, el ritmo, el contenido, la densidad, la energía y la complejidad espectral.

Un problema importante es que muchos de estos gestos pocas veces ocurren del mismo modo, a veces ocurren con diferentes alturas, intensidades, duraciones, contenidos espectrales e incluso intensiones, a este respecto nos enfrentamos a un fenómeno multidimensional que se desenvuelve en el tiempo, que va cambiando de acuerdo al contexto y la interpretación de cada persona, contrario a la imagen que puede ser un fenómeno analizable en dos dimensiones. Por ello, al igual que en la identificación de una imagen, para el análisis sonoro es indispensable que el algoritmo de escucha y aprendizaje de máquina, sea entrenado con una base de datos que contenga una amplia cantidad de muestras sonoras representativas de aquello que queremos analizar para poder generar predicciones y clasificaciones con un índice de certeza alto.

Desde esta perspectiva no es preciso entrenar al modelo computacional con improvisaciones o composiciones completas, sino con segmentos que puedan significar

una unidad gestual coherente. Al hablar de gestualidad coherente me refiero a tomar una frase melódica (en términos tonales), un gesto que permanezca similar en su densidad, timbre, contenido armónico, espectral, volumen, entre otros durante un tiempo. También podríamos hablar de morfología o tipología en términos sonoros que estarían enfocadas en el estudio de la estructura interna sonora para diferenciar sus unidades. Cabe mencionar que en conjunción, estas unidades son capaces de formar estructuras sintagmáticas musicales. Estas pueden ser entendidas como el conjunto de sonidos ordenados en el tiempo con una misma función y sentido; la construcción de frases, periodos y motivos para finalmente, dar forma a dichas estructuras. El tema relacionado con estructuras musicales será abordado a detalle en el siguiente capítulo.

3.2.1. El objeto sonoro como criterio de segmentación

Para seguir presentando los criterios de segmentación utilizados, considero importante trazar un vínculo con el concepto del objeto sonoro de Pierre Schaeffer, ya que, debido a su definición y cómo éste es concebido, resulta una perspectiva relevante para segmentar el audio y entenderlo como una unidad sonora gestual por sí misma.

En 1948, Schaeffer planteó en su tratado de los objetos musicales el término de *música concreta*, desde el cual entiende a la música, partiendo de la abstracción de los "valores musicales que contenía en potencia", como cualquier fenómeno sonoro perceptible y reproducible a través de artefactos mecánicos analógicos o digitales (Schaeffer y de Diego, 1996a). Citando al Dr. Hugo Solís al respecto sobre el tema:

> [Schaeffer] propone un conjunto de siete categorías en tres criterios, forma, materia y variación. La forma se define como una propiedad que podría observarse si pudiéramos congelar el sonido; la materia está relacionada con la evolución del tiempo; y la variación se refiere al cambio de forma y materia. En los criterios de la materia definió masa, timbre armónico y grano; en los criterios de forma dinámica y ritmo (modulación de amplitud o frecuencia); y en los criterios de variación perfil melódico y perfil de masa.(Solís, 2006)

Posteriormente Schaffer prefirió el uso del término *músicas electro-acústica*(Schaeffer y de Diego, 1996b) debido a que por definición, toda la música es concreta según su contexto cultural. Lo que me interesa rescatar de esta perspectiva, es su trabajo de clasificación sonora partiendo del término de *objeto sonoro*, el cual es un concepto clave a través del cual se construye una fenomenología de la escucha y luego del sonido. Utilizaré la definición de Michel Chion para clarificar el concepto de objeto sonoro: "Llamamos objeto sonoro a todo fenómeno y acontecimiento sonoro percibido como un conjunto, como un todo coherente, oído mediante una escucha reducida que lo enfoque por sí mismo, independientemente de su procedencia o de su significado."(Dhomont, 1984)

El concepto del objeto sonoro es tomado en cuenta al momento de generar los segmentos de audio del corpus de improvisaciones. Lo que estoy buscando es una metodología que me permita abstraer objetos sonoros reconocibles a nivel individual independientemente de su contexto y de los elementos que lo rodean. El objeto sonoro, está ligado al gesto como unidad sonora, parte de un trazo, un esbozo o un despliegue energético inicial, para evocar sensaciones, emociones y significados sin necesariamente pertenecer a un contexto musical; el gesto es la partícula que da sentido y direccionalidad al sonido, dotándolo de atributos comunicativos; es una partícula significante, abierta a su reelaboración y recontextualización, con un carácter y una intencionalidad energética que junto con otros gestos dirigen y cohesionan una idea musical abstracta.

Desde esta perspectiva, hay una analogía interesante entre el gesto y el algoritmo dadas sus cualidades abiertas y recontextualizables. Por un lado, el algoritmo, por más pequeño que sea, es capaz de comunicar una idea y de presentar una forma de conocimiento a varios niveles de abstracción. Desde estas cualidades es posible recontextualizarlo o componer con él nuevas funciones y aplicarlo a la resolución de tareas muy diversas. De modo similar, el gesto por sí mismo no requiere de mayores explicaciones, ni definiciones específicas para presentarse como un todo coherente sin un contexto previo o futuro, ya que por sí mismo significa y produce experiencias en la percepción. De forma análoga a la composición de funciones en los algoritmos, los gestos en conjunto son capaces de formar una frase o una estructura más grande

donde sus implicaciones contextuales cambian el sentido particular de cada gesto individual.

Como he mencionado en algunos momentos, a lo largo de esta investigación las aproximaciones tecnológicas y los resultados a nivel estético se han modulado por medio de los *rodeos*[3] del ciclo que incluye a la práctica artística, la investigación y el desarrollo tecnológico. Estos rodeos me llevaron a explorar diferentes aproximaciones relacionadas con el tema de la segmentación de audio en fragmentos significativos para un contexto como la libre improvisación. Considero relevante mencionar cada una de éstas aproximaciones ya que fueron de utilidad para analizar el problema y determinar cuál aproximación resultó de mayor utilidad para segmentar el corpus de audio en gestos significativos en el contexto de la improvisación libre. A continuación explico la primer aproximación que abordé.

El primer experimento consistió en una aproximación completamente manual, esto sucedió al inicio de la investigación (cuatro años atrás). Los segmentos fueron seleccionados al escuchar atentamente diferentes grabaciones de improvisaciones libres. Posteriormente fueron clasificados manualmente en cuatro grupos, para generar una base de datos supervisada, que respondiera a la propuesta de clasificación termodinámica de Ilya Prigogine, quien considera que los sistemas termodinámicos pueden dividirse en cuatro grupos; fijos, periódicos, caóticos y complejos. Inevitablemente al emplear esta metodología comencé a construir una base de datos supervisada la cual tuvo algunas limitaciones. Al revisar detalladamente esta propuesta llegué a la conclusión de que podría ser reduccionista el tratar de clasificar toda la improvisación libre a estas cuatro categorías únicamente. Además, otra limitante sería el tiempo que tomaría construir una base de datos lo suficientemente amplia con esta metodología ya que realizar esta tarea implicaba una segmentación gestual muy minuciosa a través de software de edición de audio. Por ello, seguí investigando sobre otras formas de clasificación para crear una base de datos representativa, que no partiera de los sesgos de una clasificación basada en una teoría particular sino a través de procedimientos algorítmicos.

[3]El concepto de rodeo de Latour fue abordado en el capítulo 2 y refiere a los cambios producidos en los objetivos iniciales de un problema producto de la mediación técnica.

Cabe destacar que esta aproximación produjo los primeros experimentos performáticos que realicé entre SEALI varios ensambles de improvisadores libres; entre los que estaban los músicos (nombres) Diego Villaseñor y Jorge Berumen. Es importante señalar que parte del proceso metodológico central que seguiré elaborando en esta sección derivó de estos experimentos. Sin ellos, no hubiese llegado a los resultados del desarrollo algorítmico central. Sin embargo, para no desviar la atención de la línea de desarrollo, los detalles de esta implementación se encuentran elaborados en los anexos de esta investigación, a manera de rama o bifurcación, como una línea de desarrollo posible pero paralela.

3.2.2. Segmentación basada en un ventaneo específico

El segundo experimento que realicé consistió en cortar el audio en segmentos del mismo tamaño usando la librería *AudioSegment*[4] sobre el lenguaje de programación Python. Para esta aproximación resultó relevante contar con la misma duración temporal en los segmentos de audio de la base de datos. Esto podría, en un principio, parecer más consistente que la aproximación que presento más adelante, debido a que sería posible entrenar al modelo computacional con segmentos del mismo tamaño y misma cantidad de información.

Aunque esta aproximación podría ser de utilidad para ciertas aplicaciones, no contempla el concepto del objeto sonoro, ni el planteamiento de la gestualidad y coherencia sonora o musical en cada fragmento de audio que corta el algoritmo. De manera tal que al emplear esta aproximación para la segmentación del audio, sería común encontrar segmentos que no cumplen con el criterio de segmentación considerado. El planteamiento del objeto sonoro como unidad gestual quedaría fuera y, en su lugar, se cortaría de manera indiscriminada mediante una ventana temporal fija. La problemática de seguir con esta aproximación es la mezcla tímbrica que podría ocurrir en un mismo ejemplo de audio o si el algoritmo combina en uno de sus segmentos un fragmento de gesto con uno de silencio. De este modo terminaríamos con una mayor cantidad de ruido en la información que podría ofuscar el posterior proceso de clasificación. Sin embargo, para comprobar la hipótesis anterior, esta aproximación

[4] `https://audiosegment.readthedocs.io/en/latest/audiosegment.html`

fue parte de los experimentos realizados en esta investigación. A continuación presento el código que me permitió cortar el audio en fragmentos de igual duración:

```
from pydub import AudioSegment
import math
import os

def get_duration():
    return audio.duration_seconds

def single_split(from_min, to_min, split_filename):
    t1 = from_min * cut_size * 1000
    t2 = to_min * cut_size * 1000
    split_audio = audio[t1:t2]
    split_audio.export(folder + split_filename, format="wav")

def multiple_split(min_per_split):
    total_mins = math.ceil(get_duration() / cut_size)
    for i in range(0, total_mins, min_per_split):
        split_fn = "{:06d}".format(i) + ".wav"
        single_split(i, i+min_per_split, split_fn)
        print(str(i) + "Done")
        if i == total_mins - min_per_split:
            print("All splited successfully")

cut_size = 0.1
folder = 'audio/opt/'
filename = "05 Open Paper Tree.wav"
filepath = folder + '/' + filename
audio = AudioSegment.from_wav(filepath)

multiple_split(min_per_split=1)
```

Si bien esta perspectiva puede resultar útil para entrenar al modelo partiendo de la consistencia temporal en los segmentos de audio, el proceso experimental y la perspectiva heurística de la investigación, me llevó a explorar sobre enfoques algorítmicos automatizados que tomaran en cuenta el concepto del objeto sonoro. Esto me llevó a explorar dos propuestas algorítmicas diferentes para la segmentación del corpus

de audio: *Predominant Pitch Melodia* y *SBic* ambos parte de la librería Essentia. A continuación presento estos dos desarrollos:

3.2.3. Segmentación basada en detección melódica

En el primer caso se trata de un algoritmo que detecta cambios melódicos partiendo de una serie de parámetros y compuertas destinados a la detección de cambios significativos en varios parámetros del audio a analizar. Su utilización resultó interesante debido a que cuando no logra detectar una frecuencia clara, el algoritmo regresa ceros y cuando logra detectar una frecuencia regresa el valor de la misma. Esto es de utilidad para no partir de una perspectiva de segmentación basada en la detección de momentos de inicio (Onsets) sino que el contorno armónico y melódico es tomado en cuenta para realizar esta tarea. Si bien podría tener ciertas limitaciones para detectar una frase en lenguajes no tonales ni armónicos como los producidos en la libre improvisación. Esta limitante se compensa con la amplia cantidad de parámetros que ofrece el algoritmo como la tolerancia al cambio entre las frecuencias detectadas, el número de armónicos o el peso armónico a considerar. Además, al hacer los cortes el algoritmo tomó como parte del segmento los momentos en que deja de detectar una altura, esto podría ser de ayuda en momentos donde un sonido es seguido de su resonancia o de silencio ya que genera cierta coherencia gestual al terminar el segmento justo donde comienza un nuevo tono. Por otro lado, también podría ser una limitante a la hora de clasificar debido a que el algoritmo destinado a esta tarea consideraría los segmentos sonoros y los momentos de resonancia y silencio como parte de una misma unidad. Esto podría resultar en una generalización que no necesariamente responde a la unidad gestual que estoy buscando sino que cada segmento tendría un elemento gestual sumado al silencio que le sigue. A continuación presento cuatro ejemplos del funcionamiento de la segmentación basada en el contorno melódico y el código empleado para realizar esta tarea. En las figuras 3.3 a la 3.6 podemos observar que el algoritmo de segmentación basada en el contorno melódico predominante detecta cierta altura y posteriormente baja a 0 cuando hay una ausencia de altura. Como he mencionado, para el algoritmo, ésta sección es considerada como parte del mismo segmento, y siguiendo de esta forma, cuando detecta una nueva altura la considera como un nuevo segmento.

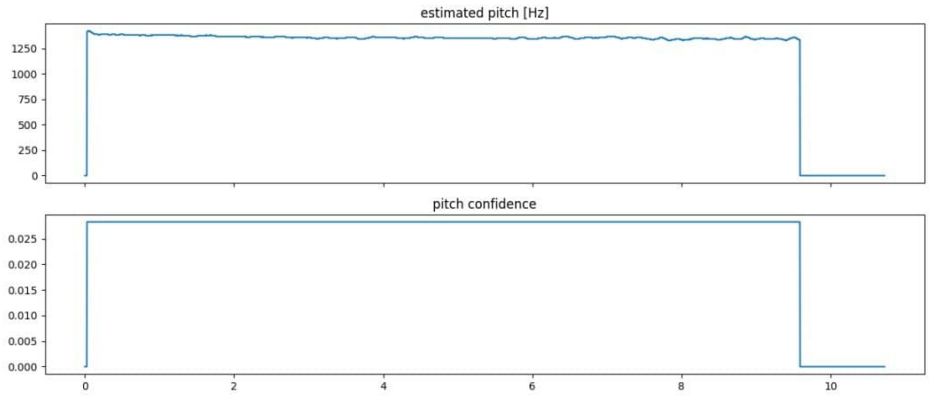

Figura 3.3: Detección del contorno melódico de un silbido. El eje de las x representa tiempo en segundos, el eje de las y es frecuencia y certeza en la detección de la frecuencia, respectivamente.

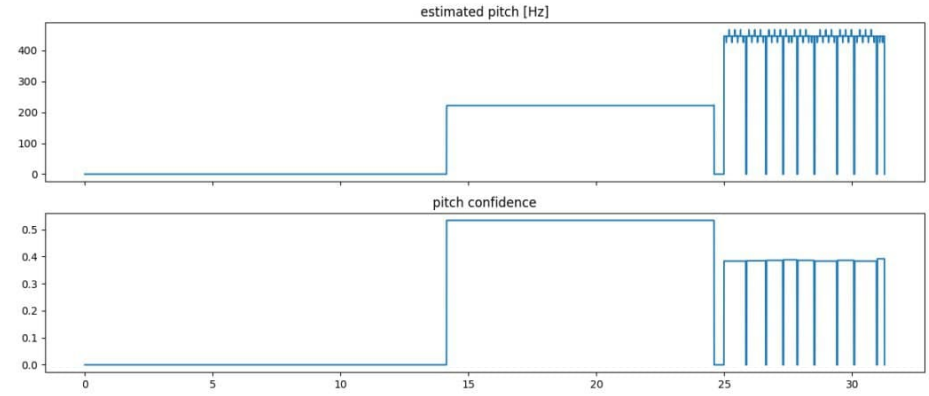

Figura 3.4: Detección de silencio, una sinusoidal a 221 Hz y un sonido complejo de dos sinusoidales a 441+450 Hz.

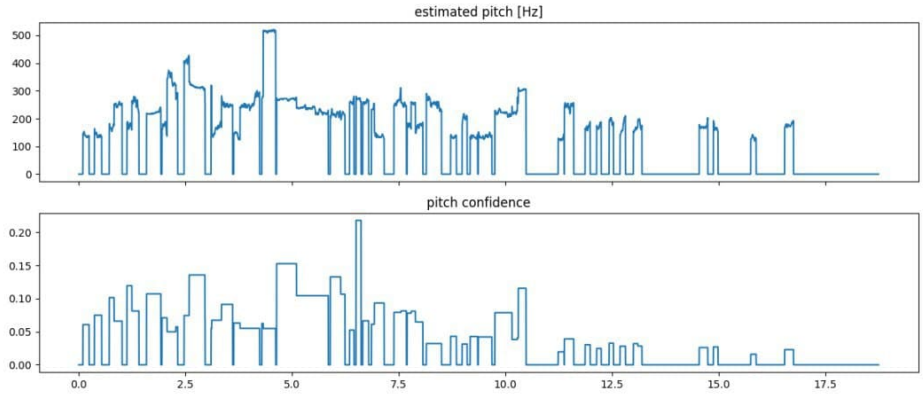

Figura 3.5: Detección melódica de *Sink Into Return* de Clare Cooper.

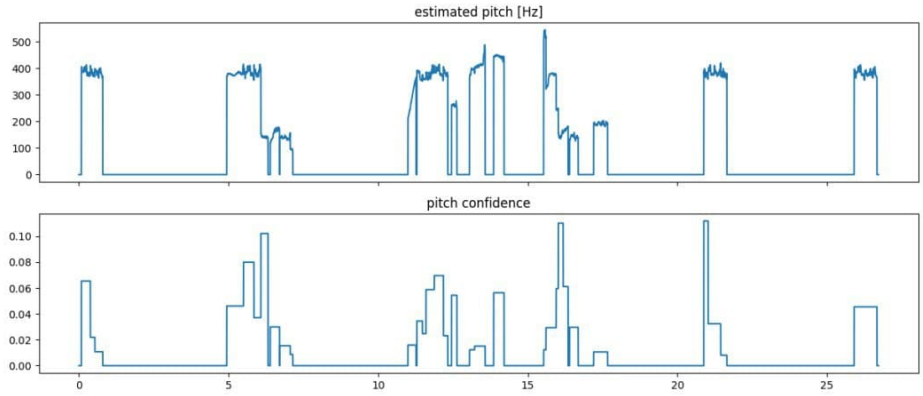

Figura 3.6: Detección melódica de una improvisación vocal y percusión de autoría propia.

```
sample_rate = 44100
hS = 512
fS = 2048

def audio_segments_generator(audio):
    pitch_extractor = PredominantPitchMelodia(frameSize=fS,
                                              hopSize=512,
                                              filterIterations=3,
                                              harmonicWeight=0.8,
                                              magnitudeThreshold=40,
                                              guessUnvoiced=False,
                                              minDuration=10,
                                              numberHarmonics=20,
                                              timeContinuity=100,
                                              voiceVibrato=False,
                                              voicingTolerance=0.8,
                                              pitchContinuity=27.5625,
                                              peakFrameThreshold=0.7)
    pitch_values, _ = pitch_extractor(audio)
    prev_index = 0
    sample_indexes = []

    for i in range(len(pitch_values)):
        num = pitch_values[i]
        if num > 0 and prev_index == 0:
            sample_indexes.append(i)
        prev_index = num

    sample_list = [n*hS for n in sample_indexes]
    audioSize = audio.size
    sample_list.insert(0, 0)
    sample_list.append(audioSize)
    return sample_list

def concat_audio(out_dir, frame_count, segment_count, segment_list,
    audio_list, file_index, output_data):
    if not os.path.exists(out_dir):
        os.makedirs(out_dir)
    print("frame, segment", frame_count, segment_count, file_index)
```

```
39      if frame_count < len(segment_list[segment_count])-1:
40          startFrame = segment_list[segment_count][frame_count]
41          endFrame = segment_list[segment_count][frame_count + 1]
42          filename = os.path.abspath(
43              os.path.join(out_dir, str(file_index)+".wav"))
44          length = endFrame - startFrame
45          if length > 0:
46              output_data.append(
47                  {"index": file_index,
48                   "path": filename,
49                   "start": startFrame,
50                   "end": endFrame,
51                   "length": length,
52                   "length-seconds": (endFrame - startFrame)/sample_rate
    ,
53                   "sample_rate": sample_rate})
54              MonoWriter(filename=filename,  format='wav', sampleRate=
    sample_rate)(
55                  audio_list[segment_count][startFrame:endFrame])
56          concat_audio(out_dir, frame_count+1, segment_count,
57                       segment_list, audio_list, file_index+1,
    output_data)
58      else:
59          if segment_count < len(segment_list)-1:
60              concat_audio(out_dir, 0, segment_count+1, segment_list,
61                           audio_list, file_index, output_data)
62      return output_data
63
64  def process_file(out_dir, filename):
65      allSampleList = []
66      allAudio = []
67      audio = MonoLoader(filename=filename)()
68      sampleSegments = audio_segments_generator(audio)
69      allSampleList.append(sampleSegments)
70      allAudio.append(audio)
71      files = concat_audio(out_dir, 0, 0, allSampleList, allAudio, 0,
    [])
72      return files
73
```

```
74 process_file('Segments', 'audio/milo.wav')
```

3.2.4. Segmentación basada en cambios de *MFCCs* mediante *SBIC*

La tercera aproximación involucró el empleo del algoritmo *SBic* ((Bayesian Information Criterion)) el cual es un algoritmo enfocado en la segmentación de audio que se encarga de diferenciar, mediante un umbral fundamentalmente espectral y otros parámetros, los segmentos que contiene una cadena de audio. *SBic* emplea un criterio de segmentación bayesiano dada una matriz de descriptores de audio. En este caso utilicé el descriptor de audio *MFCCs* (Mel-frequency cepstral coefficients)[5] como matriz de descriptores para establecer el criterio de segmentación el cual busca segmentos homogéneos que tengan una descripción semejante. El proceso de segmentación ocurre en tres fases: segmentación en bruto, segmentación fina y validación de los segmentos.

La aproximación puede estar mediada por una matriz de diferentes características de la señal digital del audio. Para este caso en concreto, debido a que la improvisación libre parte generalmente de una aproximación predominantemente tímbrica utilicé una descripción enfocada en el análisis de este parámetro, a través del descriptor de audio *MFCCs*. De manera tal que, cambios tímbricos sustanciales en audio fueran el criterio de segmentación. Esta decisión fue tomada a través de una serie de exploraciones que realicé con diferentes descriptores de audio y corroborando, a través de mi escucha, de qué manera se obtenían segmentos más concretos en los que pudiera percibir una coherencia gestual. Al final, al utilizar el gran corpus de improvisaciones, terminé con una gran cantidad de segmentos de audio que van de un centésimo de segundo hasta los dos minutos de duración. A continuación presento

[5]Los MFCC (Coeficientes Cepstrales de las frecuencias de Mel) son coeficientes inicialmente desarrollados para la representación numérica de la voz, basados en amplios estudios sobre la percepción auditiva humana. Los *MFCCs* describen las características de la señal de audio, generalmente de la voz, asociadas a las particularidades del tracto vocal. Los coeficientes cepstrales se derivan de la transformada del coseno discreta (*DCT - Discrete Cosine Transform*). Una de las particularidades de este descriptor es que las bandas de análisis de frecuencia están situadas logarítmicamente, según la escala Mel que representa la forma en la que escuchamos los humanos. Esta escala consiste en hacer un mapeo entre las frecuencias y las alturas percibidas de acuerdo al sistema auditivo humano que no percibe de manera lineal sino aparentemente logarítmica. Para una explicación más detallada sobre el funcionamiento y las particularidades de este descriptor de audio recomiendo consultar el apartado 3.1.1 *MFCCs descriptor de audio para el reconocimiento de texturas sonoras complejas* de mi tesis de maestría(Escobar Castañeda, 2016).

el código para realizar la segmentación gestual basada en el criterio de información bayesiana tomando los *MFCCs* como referencia:

```
in_dir = 'audio/provenance/'
out_dir = 'provenance_segments/'
if not os.path.exists(out_dir):
    os.makedirs(out_dir)

counter = 0

def segments_gen(fileName):
    loader = essentia.standard.MonoLoader(filename=fileName)
    audio = loader()
    w = Windowing(type = 'hann')
    spectrum = Spectrum()
    mfcc = MFCC()

    logNorm = UnaryOperator(type='log')
    pool = essentia.Pool()
    for frame in FrameGenerator(audio, frameSize = 2048,  hopSize =
    512, startFromZero=True):
        mfcc_bands, mfcc_coeffs = mfcc(spectrum(w(frame)))
        pool.add('lowlevel.mfcc', mfcc_coeffs)

    minimumSegmentsLength = 1
    size1 = 20
    inc1 = 20
    size2 = 20
    inc2 = 20
    cpw = 1

    features = [val for val in pool['lowlevel.mfcc'].transpose()]
    sbic = SBic(size1=size1, inc1=inc1,size2=size2, inc2=inc2,cpw=cpw,
     minLength=minimumSegmentsLength)
    segments = sbic(np.array(features))
    record_segments(audio,segments)

def record_segments(audio, segments):
    for segment_index in range(len(segments) - 1):
        global counter
```

```
36          start_position = int(segments[segment_index] * 512)
37          end_position = int(segments[segment_index + 1] * 512)
38          writer = essentia.standard.MonoWriter(filename=out_dir + "{:06
    d}".format(counter) + ".wav", format="wav")(audio[start_position:
    end_position])
39          counter = counter + 1
40
41  def gen_all_segments(audio_files):
42      return list(map(segments_gen, audio_files))
43
44  input_data = gen_all_segments(sorted(glob.glob(in_dir + "*.mp3")))
```

Mediante esta aproximación fue posible encontrar cierta equivalencia con el concepto de objeto sonoro siendo que las características individuales de cada segmento contienen todo el material necesario para poder identificar, desde la escucha, cada segmento como una unidad coherente a nivel gestual.

3.3. Descriptores para el análisis digital de la señal de audio

El objetivo de este apartado es construir una base de datos derivada del corpus de segmentos donde cada elemento contenga una descripción numérica que lo defina concretamente. Para ello, es preciso analizar cada uno de los segmentos y extraer sus características, esto lo realicé mediante dos aproximaciones diferentes. En la primera utilicé una librería desarrollada para el entorno de programación Supercollider Music Information Retrieval (SCMIR) desarrollada por Nick Collins y en la segunda una librería llamada *Essentia* desarrollada por el Music Tecnology Group de la Universitat Pompeu Fabra. Esta última librería fue desarrollada en el lenguaje de programación *C* y es extensible para varios lenguajes de programación. Cada una de las descripciones numéricas del audio realizadas en esta sección están escritas en archivos con extensión *CSV*. Este tipo de formato representa una de las tantas formas para conformar una base de datos y fue utilizado por su amplia y sencilla aplicabilidad en el entorno de programación Python.

3.3.1. Extracción de características con SCMIR

Supercollider Music Information Retrival[6] es una librería para SuperCollider creada por Nick Collins (Collins, 2011) enfocada en el análisis en tiempo–real de señales de audio digitales. Algunos de los descriptores de audio con los que trabaja son: *MFCC, Loudness, SpecCentroid, SpecPcile, SpecFlatness, FFTCrest, FFTSpread, FFTSlope, SensoryDissonance, Chromagram*, entre otros. Vinet *et al.* (2002)

Cabe destacar que la extracción de características con esta aproximación regresa el promedio de cada uno de los descriptores de los audios analizados (aunque puede ser configurada para realizar esta operación dado un ventaneo específico), con lo cual extrae la misma cantidad de datos (por ejemplo, 13 *MFCCs*) de todos los archivos de audio independientemente de su duración. Esta estructuración de los datos extraídos es necesaria para llevar a cabo el aprendizaje de máquina utilizado en esta investigación.

En esta etapa de la investigación he realizado muchas exploraciones combinando distintos descriptores de audio y corroborando cuál puede ser la combinación más adecuada para clasificar los segmentos. Esta tarea debería contemplar las propiedades sonoras de la improvisación libre. Si bien podríamos usar uno o varios descriptores de audio por cada una de las propiedades sonoras, los resultados de los experimentos realizados sugieren que no es lo más adecuado ya que al momento de clasificar los elementos, se producen ciertas discrepancias que tienden a producir resultados deficientes en la clasificación.[7] Debido a esto, los descriptores que consideré para realizar esta tarea fueron: *Spectral Flatness* que dada una cadena *FFT* calcula en un rango de entre 0 y 1 el porcentaje de ruido o pureza que tiene el sonido; *Chromagram* el cual entrega un conjunto de 12 valores, que representan la integración, independientemente del registro, de las 12 alturas de la escala cromática e indica dónde está mayormente depositada la energía; y *Spectral Percentile* el cual calcula dónde está depositada

[6] `https://composerprogrammer.com/code.html` Fecha de consulta 18 de Octubre 2021.

[7] Para esta tarea, también existen varios algoritmos enfocados en la reducción de dimensiones de los datos como por ejemplo *Principal Component Analysis - PCA* y *Linear Discriminant Analysis - LDA* los cuales son abordados en la sección de clasificación de este capítulo.

la distribución del espectro de frecuencias y regresa el valor de las frecuencias que corresponden al percentil deseado.[8]

//función para extraer las características de audio con SCMIR en Supercollider

```
~getAudioFeatures = {|sources, classNames = ([\unknown]), features,
    windowing, window|
    var scmirs = sources.inject(
        Dictionary.new,
        {|dict, array, index|
            var numFeatures;
            var dataArr = array.collect{|filename|
                var data;
                var file = SCMIRAudioFile(filename, features);
                numFeatures = file.numfeatures;
                file.extractFeatures();
        file.gatherFeaturesBySegments([0.0]);
        data = file.featuredata.asArray;
                data;
            };
            dict.put(
                classNames[index].debug("className"),
                dataArr.collect({|data| windowing.(numFeatures, data)
    }));
            dict;
    });
    scmirs;
};
```

3.3.2. Extracción de características con Essentia

Essentia cuenta con un gran número de descriptores de audio lo que amplió las posibilidades de combinatoria entre ellos. Para esta tarea probe con todas las combinaciones posibles entre los siguientes descriptores *MFCCs, Onset Detection, Loud-*

[8]Un resultado a considerar en estos experimentos, es que al agregar los descriptores *MFCCs*, la clasificación cambia considerablemente produciendo resultados mucho menos diferenciables entre las clases propuestas por el algoritmo.

*ness, Spectral Centroid, Spectral Flatness, Spectral Constrast, Dynamic Complexity, Spectral Complexity, Mel Bands y Chromagram*Vinet *et al.* (2002).

Al Modificar los parámetros de los descriptores de audio de essentia y probar diferentes combinaciones con ellos, obtuve resultados de clasificación muy diversos, en algunos casos poco claros al momento de clasificar las improvisaciones. En algunos casos parecía que la clasificación se hubiera hecho de forma aleatoria. Un experimento interesante a este respeto fue comparar entre una clasificación generada usando estos descriptores, otra generada por un humano y otra generada de manera aleatoria, las diferencias entre la aleatoria y la producida al usar algunas combinaciones de estos descriptores fue casi imperceptible.

Después de muchas pruebas que consistieron en escuchar, corregir y configurar los algoritmos, los descriptores que resultaron más relevantes para la clasificación fueron; *flatness, loudness, espectral centroid* y *MFCCs*; 16 descripciones en total. A continuación muestro el código para la extracción de múltiples características de audio utilizando la librería Essentia y el lenguaje de programación Python:

Extracción de multi-características de audio con Essentia

```
def extract_mfccs(audio_file):
    loader = essentia.standard.MonoLoader(filename=audio_file)
    audio = loader()
    w = Windowing(type='hann')
    fft = FFT()

    name = audio_file.split('/')[1].split('.')[-1]

    pool = essentia.Pool()
    for frame in ess.FrameGenerator(audio, frameSize=2048, hopSize
    =2048, startFromZero=True):
        mag, phase, = CartesianToPolar()(fft(w(frame)))
        mfcc_bands, mfcc_coeffs = MFCC(numberCoefficients=13)(mag)
        flatness = Flatness()(mag)
        centroid = Centroid()(mag)
        loudness = Loudness()(mag)

        pool.add('lowlevel.mfcc', mfcc_coeffs)
```

```
        pool.add('lowlevel.loudness', [loudness])
        pool.add('lowlevel.flatness', [flatness])
        pool.add('lowlevel.centroid', [centroid])

    pool.add('audio_file', (name))
    aggrPool = PoolAggregator(defaultStats=['mean','var'])(pool)

    YamlOutput(filename='features.json', format='json',
               writeVersion=False)(aggrPool)

    json_data = get_json("features.json")

    return {"file": json_data['audio_file'],
            "mfccMean": json_data['lowlevel']['mfcc']['mean'],
            "loudness": json_data['lowlevel']['loudness']['mean'],
            "flatness": json_data['lowlevel']['flatness']['mean'],
            "centroid": json_data['lowlevel']['centroid']['mean']
            }

def extract_all_mfccs(audio_files):
    return list(map(extract_mfccs, audio_files))

def getProps(props, dict):
    return map(lambda prop: dict[prop], props)

def concat_features(input_data):
    features = list(map(lambda data:
               list(tz.concat(getProps(
                   ['flatness', 'loudness', 'centroid', 'mfccMean'],
                   data))),
    input_data))
    return features

def save_as_matrix(features):
    save_descriptors_as_matrix('datasets/
    opt_flatness_loudness_centroid_mfcc.csv', features)

input_data = extract_all_mfccs(sorted(glob.glob('segments/opt_segments
    /' + "*.wav")[0:]))
```

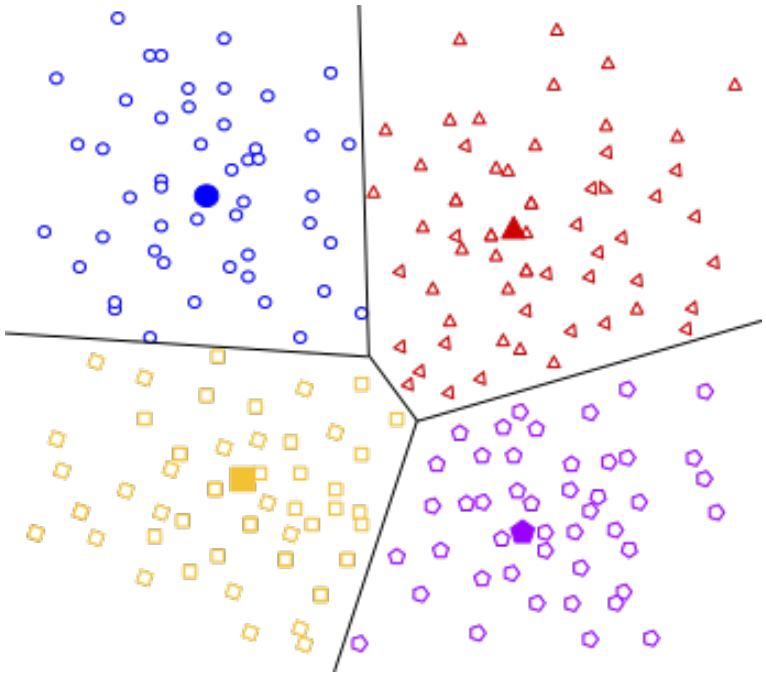

Figura 3.7: Agrupamiento de datos con el algoritmo sin supervisión *Mean Shift Clustering*

```
55 save_as_matrix(concat_features(input_data))
```

3.4. Clasificación

En el aprendizaje automático la clasificación está considerada como un caso de aprendizaje supervisado donde se encuentra disponible un conjunto de entrenamiento de datos previamente identificados o etiquetados ya sea por humanos o por un modelo de agrupamiento de datos. (Rebala *et al.*, 2019) Por ejemplo, el aprendizaje no supervisado es conocido como *clustering*, éste implica agrupar los datos en categorías que comparten ciertas características dentro de una medida de similitud. Asimismo, un modelo de agrupamiento puede entenderse como una división en regiones correspondientes donde los datos de entrada serán separados por un límite de decisión. El modelo computacional se crea a partir del conjunto de datos de entrenamiento y su utilidad es identificar la categoría a la que pertenece un elemento con características específicas. Cuando a un modelo que clasifique partiendo de un límite de decisión le es presentado un nuevo ejemplo sin clasificar, determina una etiqueta en relación con la posición relativa dentro de ese límite fijado.

El objetivo de la clasificación es identificar a qué categoría pertenece un conjunto de subcategorías (subpoblaciones) entre un conjunto de categorías previamente mapeadas. El conjunto de categorías debe estar etiquetado ya sea mediante una clasificación binaria o por una clasificación de múltiples clases. Normalmente estos conjuntos se analizan individualmente al extraer una serie de propiedades cuantificables, en este contexto se llaman "características" o "descriptores" que pueden ser números enteros o decimales (en caso de que fueran palabras habría que convertirlas a alguna descripción numérica). Los modelos de clasificación son creados al emplear umbrales simples, técnicas de regresión o técnicas de redes neuronales. Lo que se espera del clasificador es una salida (puede ser de una sola variable o multivariable) que corresponda con la distinción de categorías etiquetadas de los datos de entrenamiento. Así el modelo puede identificar elementos nunca antes vistos ya que comparten ciertas características con los que ya conoce. Otra aproximación es entrenar al modelo para predecir valores futuros dada una secuencia de datos de entrada con base en sus datos históricos, esto a través de las redes neuronales recurrentes.(Alpaydin, 2004)

3.4.1. Creación de base de datos anotada con *Mean Shift Clustering*

Una de las razones por la cual opté por trabajar con el algoritmo *Mean Shift Clustering - MSC* fue apelar al propio sentido emergente (condicionado por los sesgos entrelazados de los diferentes algoritmos empleados), que en interacción conjunta aportan los algoritmos al proceso. De forma tal, que la aproximación no partiera de una caracterización tendenciosa, dirigida por alguna una inclinación subjetiva o sesgada por grupos definidos, ya sea dentro de una teoría vinculada con la clasificación sonora o musical, sino tratara de generar un proceso que pueda prescindir, *a grosso modo*, de la toma de decisión humana para distinguir entre las diferentes categorías sonoras que pudieran presentarse en este corpus.

Una de las propuestas que he trabajado desde la maestría ha sido emplear las distancias de similaridad o diferencias sonoras para definir estos grupos en términos tímbricos. Estas diferencias pueden ser medidas por un clasificador sin supervisión como el algoritmo *Mean Shift Clustering* el cual se encargaría de agrupar los segmen-

tos sonoros detectados en diferentes clases. Una particularidad de este algoritmo es que no necesariamente requiere una indicación explícita sobre el número de grupos a generar, como es el caso del algoritmo *K-Means*, sino que trata de encontrar la mayor concentración de elementos en el espacio y actualiza los puntos centrales a la media de los elementos encontrados con base en un umbral de decisión. Posteriormente los puntos son filtrados para eliminar posibles duplicados cercanos y de este modo se forma el último conjunto de puntos centrales rodeados por los elementos cercanos a cada conjunto.

El primer experimento que realice a este respecto fue analizar las grabaciones de dos improvisaciones libres; *Sink Into Return* de Clare Cooper y *Open Paper Tree* de Michael Doneda, Paul Rogers y Le Quan Ninh, utilizando los cuatro descriptores antes mencionados (*flatness, loudness, centroid* y *MFCCs*, con 16 vectores en total, dado que el *MFCCs* tiene 13 vectores). El algoritmo pudo discernir entre elementos rítmicos, percutidos, periódicos, sonoridades superpuestas (interacción entre varios músicos), aislamiento de timbres, diferenciación entre elementos extremadamente ruidosos o casi silentes, resonancias y densidad sonora, además de avizorar entre elementos que tienden hacia sonoridades largas distinguiendo entre lo grabe y lo agudo. Cabe mencionar que la clasificación no es cien por ciento certera y que en casos como *Open Paper Tree* sucede que el algoritmo de clasificación decide mezclar en una de sus clases un segmento de flauta y tam-tam frotado con algún objeto metálico junto con otro segmento que contiene los aplausos del final de la improvisación dejando de lado otros segmentos con aplausos apartados en una clase diferente.

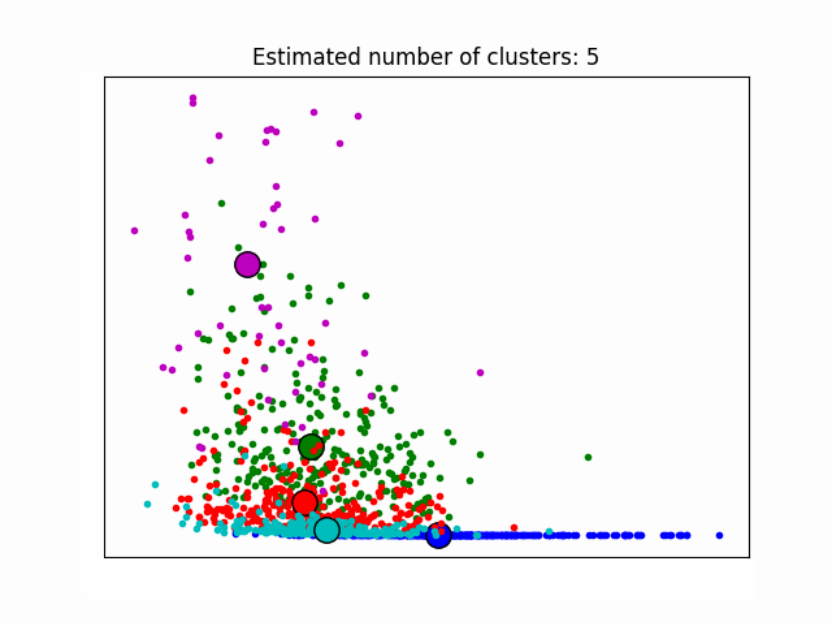

Figura 3.8: Segmentos de *Sink Into Return* agrupados por el algoritmo *Mean Shift Clustering*. En el eje de las "x" se encuentran las clases propuestas por el algoritmo, en el eje de las "y" los promedios de valores de las descripciones numéricas.

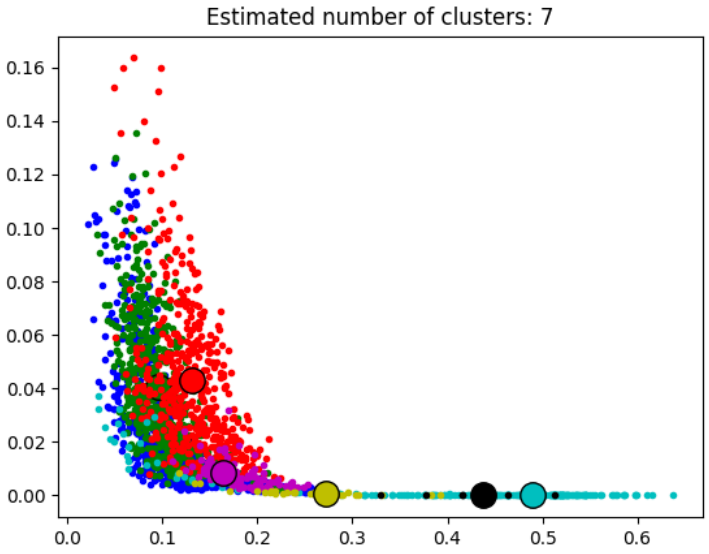

Figura 3.9: Segmentos de *Open Paper Tree* agrupados por el algoritmo *Mean Shift Clustering*. En el eje de las "x" se encuentran las clases propuestas por el algoritmo, en el eje de las "y" los promedios de valores de las descripciones numéricas. Los círculos grandes indican el centro de cada uno de los clusters.

En un segundo experimento, opté por aplicar un escalador estándar a los datos, utilizando los mismos descriptores (*flatness, loudness, centroid* y *MFCCs*), estos fueron los resultados de clasificación:

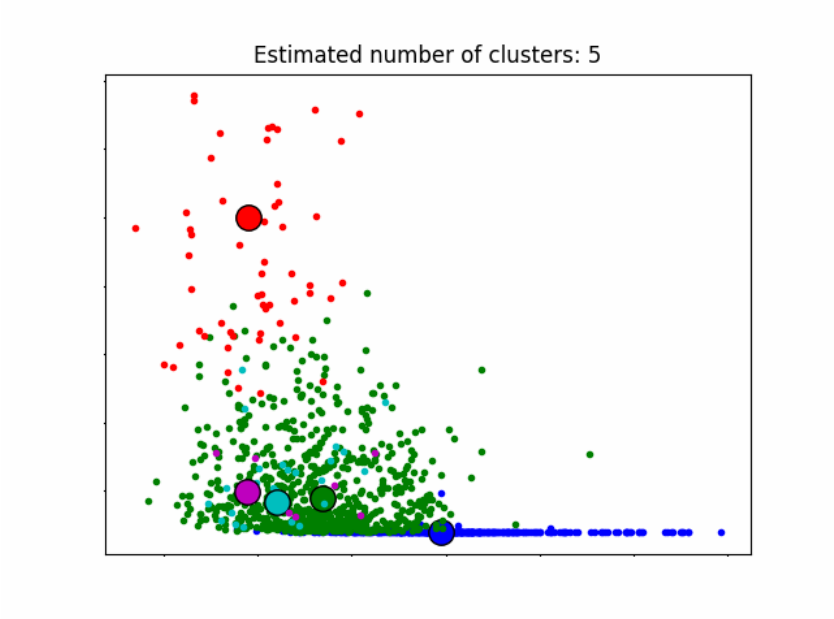

Figura 3.10: Sink Into Return - MSC Standard Scaler

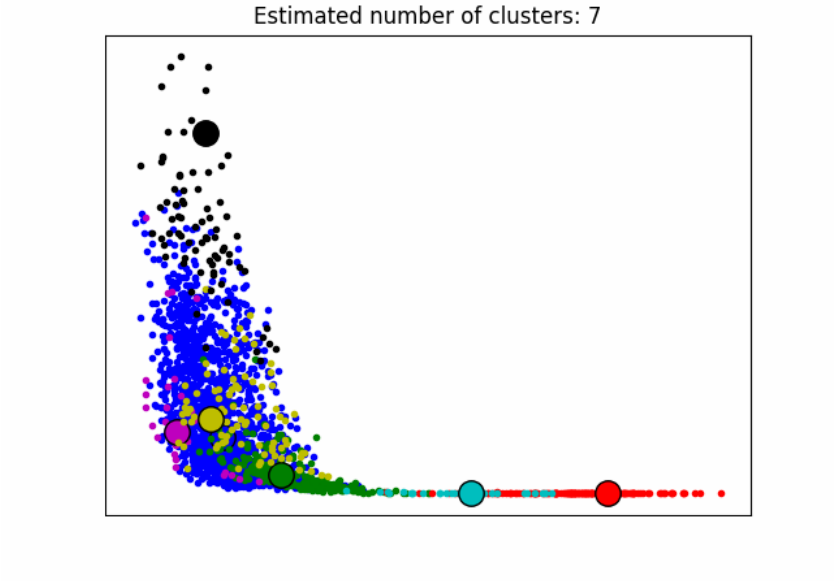

Figura 3.11: Sink Into Return - MSC Standard Scaler

Para continuar, opté por escalar (a un rango de valores de 0 a 1) la base de datos de los segmentos y aplicarle un PCA [9] tomando solo los 2 elementos más relevantes de los descriptores. Al emplear esta aproximación para clasificar el audio resulta una organización muy compleja que contempla algunos de los elementos que estoy buscando resaltar. Al procesar previamente estas grabaciones con el algoritmo *Equal Loudness*, el cual genera una normalización en los archivos de audio y extraer sus características utilizando cuatro descriptores (*flatness, loudness, centroid* y *MFCCs*), el algoritmo clasificador generó resultados, a nivel del ordenamiento del audio por clases, muy similares que sin usar el escalador, aunque la distribución visual de los elementos en las clases fue diferente. A continuación muestro estos resultados:

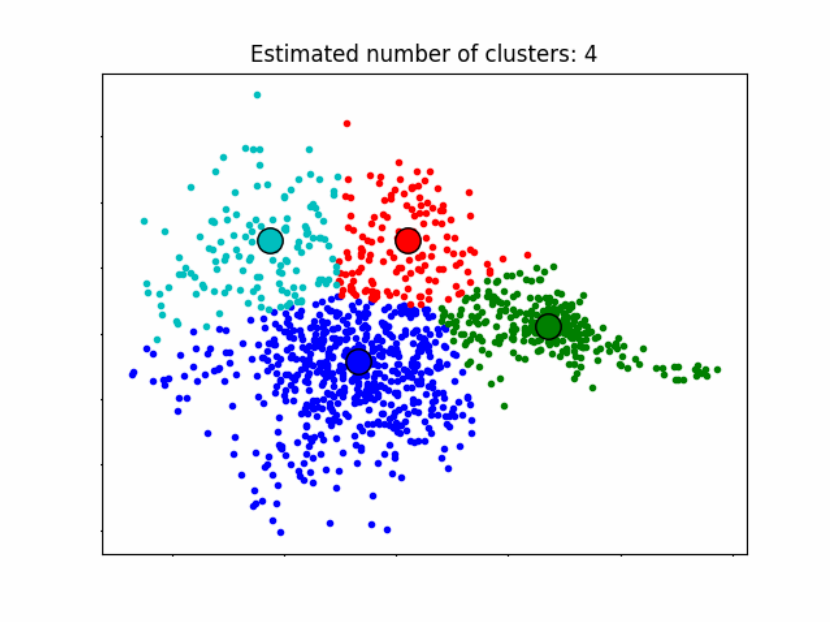

Figura 3.12: *Sink Into Return - MSC + Standard Scaler + PCA 2*

[9]Principal Component analysis for Machine Learning PCA: algoritmo que recoge los datos más relevantes dentro de un conjunto de datos.

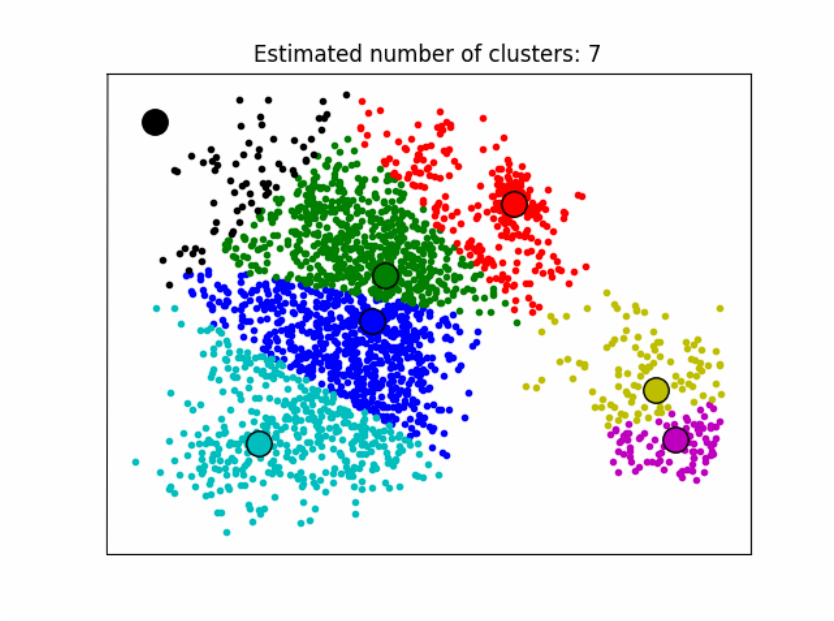

Figura 3.13: *Open Paper Tree - MSC + Standard Scaler + PCA 2*

Al hacer el mismo experimento anterior, y tomar los 10 elementos más relevantes de la distribución que consideró el algoritmo del PCA, ocurrió algo incluso mucho menos diferenciable en términos de clasificación que solo al usar 2 elementos distinguidos por el PCA.

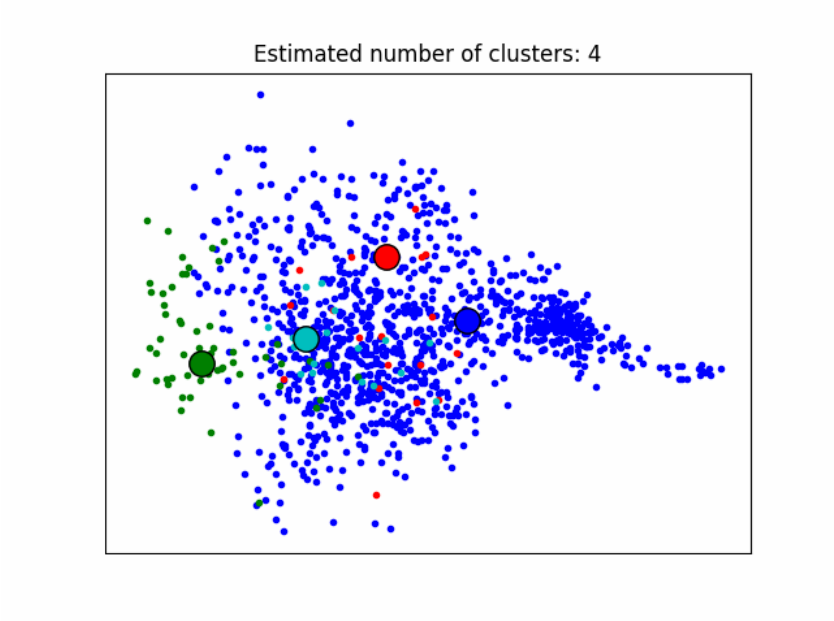

Figura 3.14: *Sink Into Return - MSC + Standard Scaler + PCA 10*

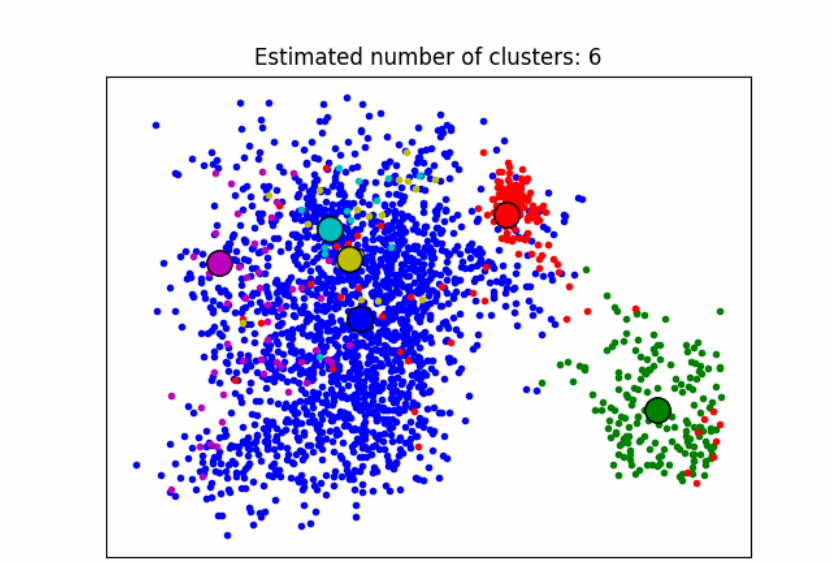

Figura 3.15: *Open Paper Tree - MSC + Standard Scaler + PCA 10*

Posteriormente realicé un último experimento con *Open Paper Tree*, cambiando considerablemente algunos de los descriptores del audio a analizar.

Los resultados fueron satisfactorios en cuanto a la clasificación no supervisada con *MSC*. Al emplear 13 y 40 coeficientes *MFCCs* generó 9 clases donde percibo cierta consistencia en cuanto a la distribución de los materiales sonoros elegidos. Se distingue cierta claridad en la distribución tímbrica; deja aislada la percusión cuando está sola, esto es muy claro en la clase 6. Aunque en la clase 7 puso el ejemplo de los aplausos junto con un ejemplo donde hay una flauta y un platillo. El silencio lo deja solo. En la clase 3 puso sonidos mínimos con volumen bajo y resonancias alargadas.

A qué se deben estás decisiones, qué hace similar el primer segmento de flauta y tam-tam con el segmento de los aplausos finales de la improvisación, sería pertinente dejar así la clasificación, qué implicaciones conlleva dejar pasar esta forma de clasificar que para nosotros podría ser un error pero que para el sistema es algo que está relacionado a través de las similaridades numéricas que hay entre los segmentos clasificados. A este respecto, sería un error pensar que la escucha de máquina se asemeja a la forma en que los humanos recibimos y comprendemos a varios niveles

los fenómenos sonoros, y que a pesar de que muchas de estas herramientas fueron creadas con base en estudios acústicos, psicoacústicos y fisiológicos sobre la percepción sonora humana, sería muy difícil hablar de un símil entre percepción escucha y entendimiento humano y de máquina. Desde ahí, sería más preciso hablar de que los sesgos implicados en la persona que desarrolla un sistema, los datos recogidos y en las herramientas utilizadas para resolver esta tarea, dan lugar a una escucha de máquinas que inevitable e irreductiblemente difiere de la escucha humana.

Cabe señalar que todos estos experimentos los realicé previo a probar las diferentes aproximaciones con la gran base de datos que describí en la subsección *Corpus musical y base de datos*. Si bien, estos experimentos demuestran que es posible crear bases de datos anotadas por un algoritmo de clasificación sin supervisión, al realizar el experimento con la gran base de datos, un problema a resaltar es que en la presencia de muchos más timbres de distintas naturalezas (electrónicos y acústicos), es posible observar un buen performance en algunas de las clases como las casi silentes pero hay otras clases que contenían más sonidos donde es sumamente difícil descifrar el criterio tomado por el algoritmo para hacer la clasificación de los diferentes segmentos de audio. Por consiguiente, puedo concluir que entre más elementos de diferente orden haya, será mucho más difícil que el sistema logre generar una buena diferenciación de los audios. Por tanto, se puede apreciar que si la complejidad y la cantidad de los ejemplos aumenta, la forma en la que el algoritmo de clasificación sin supervisión prosigue, puede resultar en muchos más errores (para una escucha humana) de clasificación. En ese sentido, a qué responde este funcionamiento que para una escucha entrenada o no en música podría resultar erróneo. Cómo está "escuchando" la máquina y cuales son los parámetros para tomar ciertas decisiones de clasificación. Dados estos resultados me ceñí a las siguientes adecuaciones.

Adecuaciones con base los resultados anteriores

De acuerdo con los experimentos previos y tomando en cuenta las problemáticas y aciertos, para analizar la gran base de datos la cual, como vimos en el apartado previo, fue básicamente imposible categorizar (en términos de que el clasificador pudiera sugerir una agrupación que fuera estéticamente diferenciable por un humano), debi-

do a la gran diversidad sonora en las improvisaciones, usando el algoritmo de *MSC*. La gran complejidad tímbrica, rítmica, calidades en grabación, etc. de los ejemplos, demuestra que el algoritmo clasificador *MSC* no es lo suficientemente robusto para lidiar de forma más certera, con lo que podríamos llamar una generalidad o corpus mucho más vasto sobre las propiedades gestuales de la improvisación libre. Por ello, consideré necesario reducir el campo de aplicación del algoritmo de clasificación a un contexto más acotado, como podrían ser las aproximaciones de algunos improvisadores libres hacia los instrumentos de cuerda punteada; para ello decidí delimitar el análisis a diferentes improvisadores libres que partieran de la utilización de guitarras acústicas, eléctricas, procesadas y sin procesar electrónicamente, así como ejemplos Clare Cooper tocando el arpa. Cabe mencionar que al limitar el espectro de posibilidades en la base de datos a un espacio tímbrico y aproximación estética similar en términos tímbricos, no dejamos de prescindir de una amplia gama de recursos sonoros que los improvisadores apelan dentro de su práctica. Así, el campo tampoco queda de un inicio tan restringido ya que los improvisadores, en algunas ocasiones tienden a recurrir a elementos sonoros como; chasquidos, gritos, el uso de su voz. Además de toda esa diversidad sonora, al momento de realizar una grabación pueden presentarse elementos sonoros que tal vez no fueron previstos como; respiraciones, suspiros, aplausos, el sonido de algún ave, etc.

Tomando estás consideraciones y diversidad en el panorama improvisativo de este estudio de caso, para realizar la compilación de esta base de datos contemplé, en la medida de lo posible, no tomar en cuenta las grabaciones con elementos que no fueran cercanas al empleo de guitarras, guitarras eléctricas y arpa, dado que como constaté en los anteriores experimentos, el darle una gama tan amplia de elementos sonoros, podrían hacer de la clasificación una tarea casi imposible de resolver para el algoritmo de clasificación *MSC*. A continuación presento los resultados de clasificación producidos por el algoritmo *MSC* al realizar la base de datos anotada bajo los criterios antes mencionados.

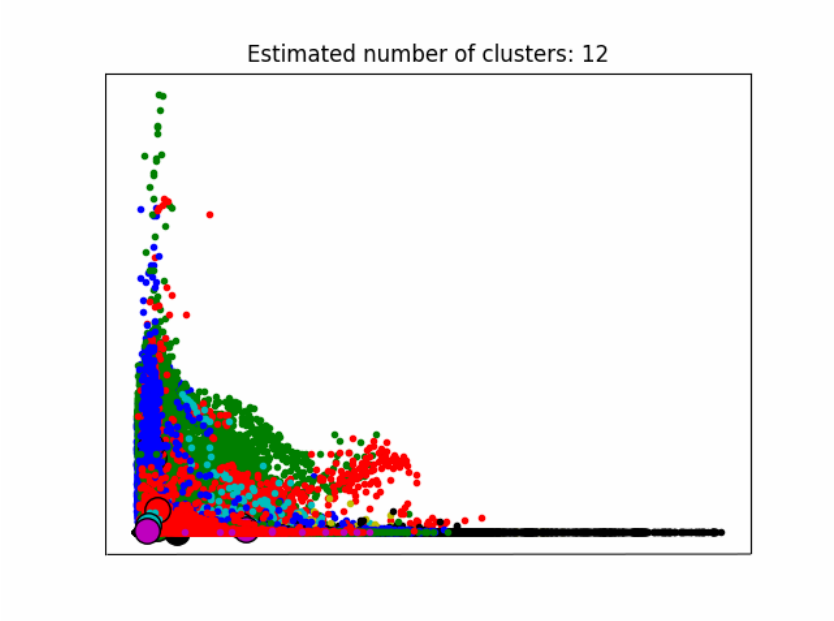

Figura 3.16: Clasificación de la base de datos de instrumentos de cuerda punteada con el algoritmo de agrupamiento *MSC*

Ahora presento el algoritmo de clasificación con el cual realicé esta tarea de clasificación:

Algoritmo de clasificación Mean Shift Clustering

```
def meanShift(features):
    tr_features = StandardScaler().fit_transform(features)
    features_pca = tr_features
    bandwidth = estimate_bandwidth(features_pca, quantile=0.021,
                                   n_samples=None)
    ms = MeanShift(
                bandwidth=bandwidth,
                bin_seeding=True,
                max_iter=100,
                cluster_all=True)
    ms.fit(features_pca)
    labels = ms.labels_
    cluster_centers = ms.cluster_centers_

    labels_unique = np.unique(labels)
    n_clusters_ = len(labels_unique)
    P = ms.predict(features_pca)
```

```
18      return n_clusters_, labels, cluster_centers, features_pca
19
20  def moveToFolders(n_clusters_, labels, folder_in, folder_out):
21      files = []
22      for audio_files in sorted(glob.glob(folder_in + "*.wav" )):
23          names = audio_files.split('/')[1]
24          files.append(names)
25
26      with open('archivos_clases.txt', 'w') as f:
27          writer = csv.writer(f, delimiter=' ')
28          writer.writerows(zip(files, labels))
29
30      clasescontent = open('archivos_clases.txt').readlines()
31      clases = [int(x.split(" ")[1]) for x in clasescontent]
32
33      for createFolders in range(n_clusters_):
34          folder = folder_out + str(createFolders)
35          if not os.path.exists(folder):
36              os.makedirs(folder)
37
38      for clase in range(n_clusters_):
39          ele = np.where(np.array(clases) == clase)[0]
40          for elements in ele:
41              num = folder_in+'{:06d}'.format(elements)
42              for audio_files in glob.glob(num + "*.wav"):
43                  shutil.copy(audio_files, folder_out + str(clase))
44
45  def assing_label_to_dataset(input_file, output_file, transform_row):
46      with open(input_file, 'r') as read_obj, \
47              open(output_file, 'w', newline='') as write_obj:
48          csv_reader = reader(read_obj)
49          csv_writer = writer(write_obj)
50          for row in csv_reader:
51              transform_row(row, csv_reader.line_num)
52              csv_writer.writerow(row)
53
54  file_in = "datasets/opt_flatness_loudness_centroid_mfcc.csv"
55  folder_in = "segments/opt_segments/"
56  folder_out = "audio/clusters_opt_nada/"
```

```
features = loadtxt(file_in)

a, b, c, d = meanShift(features)
ploter(n_clusters_=a, labels=b, cluster_centers=c, X=d)
moveToFolders(a, b, folder_in=folder_in, folder_out=folder_out)
csv_fileOut = "opt_flatness_loudness_centroid_mfcc_classes.csv"
labels = b.tolist()
assing_label_to_dataset(file_in, csv_fileOut, lambda row, line_num:
    row.append(b[line_num -1]))
```

3.5. Generación de modelos con base en la clasificación anotada

La función principal de un modelo y del aprendizaje automático es brindarnos la posibilidad de procesar, interpretar y analizar datos, sin tener que describir explícitamente las reglas que describan el problema en cuestión para desarrollar un algoritmo. En contraposición a los sistemas expertos que sí requieren una clara delimitación formal en cuanto código, los modelos construídos a partir de la utilización de redes neuronales son capaces de abstraer y sintetizar información para encontrar patrones coherentes que le permiten identificar o predecir nueva información. Estas capacidades son de gran utilidad para la resolución de problemas de diversa índole, incluidos aquellos que tienen una relación con los lados más sensibles de la naturaleza humana como son la creatividad o el arte. El modelo se construye al comunicar relaciones atendidas entre diferentes tipos o modalidades de datos. Este proceso se conoce como entrenamiento e incluye la realización iterativa de operaciones que involucran la actualización de pesos *weigths* y sesgos *bias*, la optimización y la regularización. Estos cuatro elementos proveen al modelo la capacidad de establecer espacios de límites de diversa índole que le permiten identificar a cuáles de sus categorías pertenece una nueva observación. Estas capacidades permiten a los desarrolladores usar nuevos datos para conocer comportamientos, desviaciones, tendencias u otras propiedades que el modelo podría identificar, imitar, predecir, representar, etc.

Es importante saber cuales son las propiedades de los modelos ya que de este conocimiento y de los resultados que se obtienen al finalizar con el entrenamiento es

posible realizar un control heurístico para compensar los posibles errores y acercar cada vez más los resultados arrojados por el modelo a los objetivos iniciales para la resolución del problema en cuestión. Sin embargo, generalmente

> Las personas que utilizan el aprendizaje automático en contextos musicales a menudo se preocupan por las propiedades de los modelos que no pueden medirse adecuadamente con pruebas empíricas automatizadas ni manipularse fácilmente mediante la elección del conjunto de datos de entrenamiento. A veces, el aprendizaje de algoritmos puede exponer parámetros que permiten a los usuarios un control más directo sobre las propiedades que les interesan. (Fiebrink, 2011).

Si bien este no siempre es el caso con las aplicaciones de más alto nivel para hacer aprendizaje automático, hay algunas otras que sí lo permiten en cierta medida como el software *Weka* o *Wekinator*. Aún así tienen sus límites. De este modo, usar librerías de aprendizaje automático de más bajo nivel puede ser de bastante utilidad para establecer dichos criterios heurísticos partiendo de un control total que el desarrollador pueda tener sobre la arquitectura de la red neuronal, el tipo de procesamiento en el que entrega los datos a la red y los procesos de regularización u optimización aplicados al proceso de entrenamiento. Un método heurístico bastante común entre la comunidad de músicos interesados en la utilización del aprendizaje automático para la creación musical es descrito por (Fiebrink, 2011):

> Los algoritmos de aprendizaje supervisado de propósito general para clasificación y regresión pueden entonces aprender la relación entre entradas y salidas. Los usuarios evalúan los modelos entrenados ejecutándolos en tiempo real, observando el comportamiento del sistema (por ejemplo, sonidos sintetizados) a medida que generan nuevas entradas (por ejemplo, movimientos corporales). Los usuarios también pueden modificar iterativamente los ejemplos de entrenamiento y volver a entrenar los algoritmos sobre los datos modificados. El aprendizaje automático interactivo puede permitir a los usuarios corregir fácilmente muchos errores del sistema a través de cambios en los datos de entrenamiento. Por ejemplo, si el modelo

> entrenado emite un sonido incorrecto para un gesto dado, el usuario puede grabar ejemplos adicionales de ese gesto junto con el sonido deseado y luego volver a entrenar. Esto también permite a los usuarios cambiar el comportamiento del sistema a lo largo del tiempo, por ejemplo, agregando de forma iterativa nuevas clases de gestos hasta que la precisión comienza a verse afectada o no hay necesidad de clases adicionales. El aprendizaje automático interactivo también puede permitir que las personas creen modelos precisos a partir de muy pocos ejemplos de entrenamiento: al colocar de manera iterativa nuevos ejemplos de entrenamiento en áreas del espacio de entrada que son más necesarias para mejorar el rendimiento del modelo (por ejemplo, cerca de los límites de decisión deseados entre clases).

A este respecto (Fiebrink, 2011) también menciona una característica similar al ciclo práctica artística, desarrollo tecnológico e investigación vinculado con la retroalimentación producida cuando los músicos interactúan con el modelo generado por la red neuronal:

> [La] interacción con los algoritmos puede afectar el proceso creativo de un músico de maneras que van más allá de la mera producción de modelos más precisos. Por ejemplo, el aprendizaje de máquina interactivo mediante la regresión puede convertirse en una herramienta eficaz para la exploración y para acceder a relaciones inesperadas entre las acciones humanas y las respuestas de la máquina. En el trabajo con compositores que construyen nuevos instrumentos controlados por gestos utilizando aprendizaje automático interactivo, (Fiebrink *et al.*, 2012) observaron una estrategia útil para acceder a nuevos sonidos y relaciones gesto-sonido, al tiempo que fundamentaba el diseño del instrumento en las propias ideas del compositor: los compositores primero decidieron sobre los "límites sonoros y gestuales del espacio compositivo (p. Ej., los valores máximos y mínimos de los parámetros de [síntesis] y posiciones [gestuales] del controlador", luego crearon un conjunto de datos de entrenamiento inicial empleando algunos gestos y sonidos en estos extremos. Después de entrenar las redes

> neuronales sobre estos datos, los modelos de regresión continua resultantes permitieron a los compositores moverse por el espacio de los gestos, descubriendo nuevos sonidos entre y fuera de los límites de estos ejemplos de "ancla". (Fiebrink, 2011)

De manera similar a lo descrito en los textos de (Fiebrink, 2011) y (Fiebrink *et al.*, 2012), en el caso de SEALI, una vez que la base de datos fue analizada por el algoritmo Mean Shift Clustering, que se encargó de agrupar los segmentos similares y depositarlos en diferentes carpetas dependiendo la clase que les asignó, los segmentos fueron nuevamente analizados por los mismos descriptores de audio y al finalizar, fue posible construir una base de datos anotada definida por el algoritmo de agrupamiento *MSC*. Este proceso puede ser posteriormente supervisado por una escucha humana para corroborar y/o limpiar las posibles discrepancias detectadas. A este respecto decidí mantener en cierto porcentaje la propuesta de clasificación del sistema apelando al proceso de colaboración humano-máquina, en este caso, la máquina imprime su algoritmicidad derivada del proceso para la toma de decisión enfocada en el reconocimiento de materiales sonoros.

3.6. Técnicas de interactividad algorítmica y predicción tímbrica

Para trazar las técnicas incluídas en los procesos interactivos de SEALI, parto de la búsqueda de un equilibrio entre varios de los componentes que considero importantes al momento de improvisar, algunos de ellos son: la imprevisibilidad, el azar, la contingencia, la indeterminación que surge del propio proceso de creación musical al momento y por otro lado la concepción formal, la estructura, los lugares comunes y los vaivenes de intensidad morfológica construidos. En ese sentido, la propuesta es desarrollar un sistema capaz de integrar estás perspectivas al utilizar las posibilidades y limitaciones que el mismo aparato tecnológico y necesariamente mi desarrollo como programador permiten a la hora de implementar los algoritmos necesarios para concretar los objetivos planteados. La técnica, en este sentido, limita y posibilita la dimensión conceptual y estética del proyecto. Por ejemplo, al día de hoy no podemos hablar de un sistema que sea capaz de generar audio de alta fidelidad en tiempo–real

usando algún mecanismo relacionado con redes neuronales profundas o redes neuronales recurrentes. Los ejemplos que hasta el momento he encontrado en la literatura son capaces de hacerlo con sus debidas limitaciones, o tienen tiempos de producción altos y buenos resultados sonoros aunque cortos, o tienen tiempos de procesamiento relativamente cortos (3 minutos) pero con resultados audibles de poco interés.[10] [11] [12]

Debido a las limitaciones para generar audio crudo en tiempo real, en la parte reactiva de SEALI me limito a usar puramente síntesis de sonido producida en en tiempo–real y muestras de audio que están sujetas a transformaciones de diversa índole (estiramientos temporales, síntesis granular, reverberación, delay, tratamientos de envolventes y transformación gradual de la señal), estas muestras podrían pertenecer a una base de datos previamente estructurada con índices de similaridad tímbrica o ser grabadas, analizadas y utilizadas en el mismo momento del performance.

Con la finalidad de generar un modelo computacional robusto que permita recibir nueva información e interpretarla en línea a través de la base de datos anotada, hago uso del aprendizaje supervisado partiendo de una red neuronal profunda construida con la librería TensorFlow. Esta red está compuesta por una capa de entrada con 13 neuronas y 3 capas ocultas compuestas de 4096 neuronas cada una y una capa de salida compuesta de 7 neuronas. Lo cual da un total de 13,229,959 parámetros que intervendrán en el proceso de entrenamiento del modelo. Una vez generado el modelo tiene la capacidad de hacer predicciones al vuelo de nuevos ejemplos de audio mediante Tensorflow Server.

Primer aproximación interactiva de SEALI

Una vez que el modelo ha sido generado puede ser ejecutado y hacer predicciones de nuevos ejemplos de audio en tiempo–real. Cabe mencionar que los nuevos ejemplos a analizar son capturados por un micrófono (o en su defecto mediante una conexión interna vía JackClient) los cuales están sujetos a una ventana temporal previamente asignada. Ésta puede variar dependiendo de cómo fue entrenado el modelo. En es-

[10] https://openai.com/blog/musenet/
[11] https://magenta.tensorflow.org/perceiver-ar
[12] https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio

te caso usé una duración de dos segundos, para extraer las características de cada ejemplo mediante SuperCollider y la librería SCMIR. Las características extraídas de cada segmento de audio son enviadas vía OSC a Python y la librería Request[13] los convierte a formato JSON que es una forma de enviar la información procesada a TensorFlow Sever y poder clasificarla. Una vez que la información ha sido clasificada ésta regresa a Python quien se encarga de convertirla a OSC para ser recibida por SuperCollider. Una vez que los datos de entrada de la clasificación han sido recibidos por SuperCollider son procesados por el módulo de generación sonora. Derivado de una serie de reglas compuestas por límites de certeza en la clasificación, este módulo lanza distintos archivos de audio de la base de datos clasificada los cuales están sujetos a procesos de estiramientos temporales y cambios de tono, síntesis granular, generando un continuo caudal sonoro a través de la síntesis concatenativa. Además, el módulo activa distintos sintetizadores los cuales modifican sus parámetros a partir de datos de clasificación recibidos, esto podría considerarse un proceso de sonificación de las clasificaciones del modelo.

Esta fue la primera metodología enfocada en el análisis tímbrico de improvisaciones libres y que utilicé en varios de los performances de *Resistencias Maquínicas* que son presentados en el capítulo 5. En esa sección serán discutidos a detalle y hablaré de las particularidades técnicas y estéticas de cada uno.

A continuación presento un diagrama del funcionamiento antes descrito y los códigos correspondientes a esta aproximación:

Código de Python para enviar y recibir datos a tensorflow server y Supercollider

```
1 import argparse
2 import math
3 import requests
4 from pythonosc import dispatcher
5 from pythonosc import osc_server
6 from pythonosc import udp_client
7 import json
8
```

[13]https://pypi.org/project/requests/

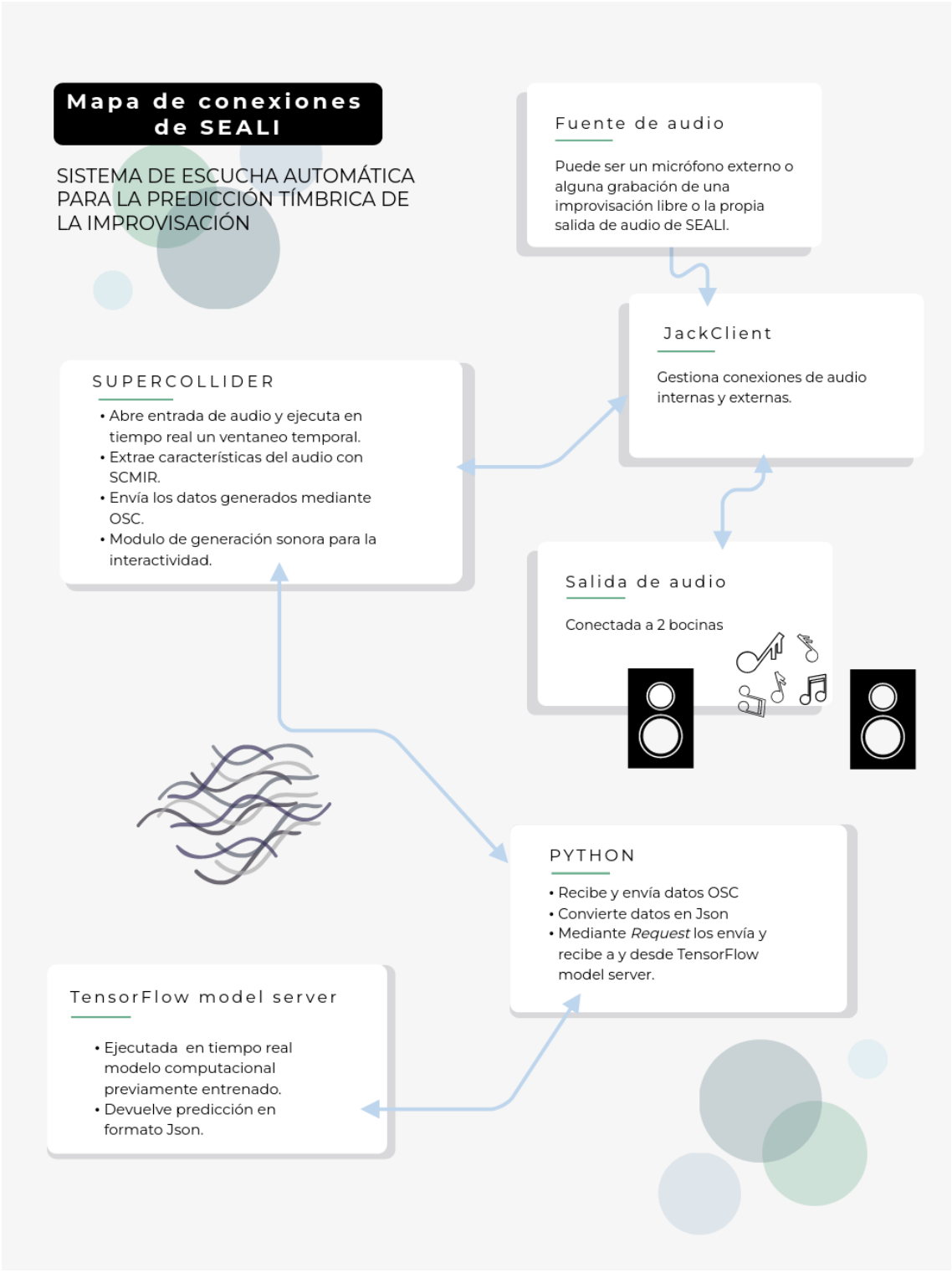

Figura 3.17: Mapa de conexiones de SEALI para la predicción tímbrica e interacción en la improvisación libre.

```
client = udp_client.SimpleUDPClient('127.0.0.1', 5006)

def tf_handler(unused_addr, *args):
  headers = {"content-type": "application/json"}
  data = {"instances": [[*args]]}
  r = requests.post(url = "http://localhost:8501/v1/models/
    improv_class:predict", data=json.dumps(data), headers=headers)
  data = r.json()["predictions"]
  client.send_message("/clase", *data)

if __name__ == "__main__":
  parser = argparse.ArgumentParser()
  parser.add_argument("--ip",
      default="127.0.0.1", help="The ip to listen on")
  parser.add_argument("--port",
      type=int, default=5005, help="The port to listen on")
  args = parser.parse_args()

  dispatcher = dispatcher.Dispatcher()
  dispatcher.map("/features", tf_handler)

  server = osc_server.ThreadingOSCUDPServer(
      (args.ip, args.port), dispatcher)
  print("Serving on {}".format(server.server_address))
  server.serve_forever()
```

Código de Supercollider que activa entrada de audio, extrae características y recibe predicciones

Cabe mencionar que aquí solo se presenta el código general, los detalles de esta implementación se pueden encontrar en el siguiente enlace de github. [14].

```

var features = [[Chromagram],[SpecPcile, 0.95],[SpecPcile, 0.80],[
    SpecFlatness],
    [OnsetStatistics]];
var window = 2;
t = ~startRec.("~/Desktop/recs", 2, {|fileName|
```

[14] https://github.com/Atsintli/SEALI/tree/master/SC_version

```
    Task({
        0.05.wait;
        ~res = ~getAudioFeatures.([[fileName]], nil, features, ~
    windoing, window);
        ~data = ~res[\unknown].flatten.flatten;
    }).play
});

~client = NetAddr("127.0.0.1", 5005); // loopback ----

Task({
    inf.do({
        ~client.sendMsg("/features", *~data);
        0.01.wait;
    });
}).play;
)
```

Código del módulo de interactividad de Supercollider

```

OSCdef (\osc2, {|msg, time, addr, recvPort|
    var clase, caotico, complejo, fijo, periodico;
    msg.debug("======");
    #clase, caotico, complejo, fijo, periodico = msg;

    //caoticos

    if((caotico < 0.0002),{
        Ndef(\sin, {
            Silent.ar
        })
    });

    if((caotico > 0.0003) && (caotico < 0.02),{
        Ndef(\buf, {
            var sig, env;
            sig = PlayBuf.ar(2, ~fijo[0..~fijo.size].choose, (-8..-1).
    choose.midiratio, 1, doneAction: 2);
            env = EnvGen.kr(Env.new([0, 0.5, 0.8, 0], [2, 30, 2]));
```

```
            sig = sig * env;
        })
    });

    if(
        (caotico > 0.03) && (caotico < 0.09), //condition
        {Ndef(\sin,{
            Resonz.ar(
                Mix.fill(28,{
                    var freq, numcps;
                    numcps= rrand(2,20);
                    Pan2.ar(Gendy4.ar(1, 100, 1.0.rand,1.0.rand
,233,235, 1, 1.0.rand, numcps, 12, 1/(28.sqrt)), 1.0.rand2)}),
MouseX.kr(233,466), MouseY.kr(0.1,4.0))
        })
    });

    if((caotico > 0.1) && (caotico < 0.2),
        {
            Ndef(\sin, {
                var in, freq, hasfreq;
                in = SoundIn.ar(0);
                #freq, hasfreq = Tartini.kr(in);
                SinOsc.ar((freq*1.1).lag(0.2)!2,0,0.2)})
    });

    if((caotico > 0.3),
        {
            Ndef (\sin, {|amp|
                //var freqs = [115, 105];
                Mix.new(Pan2.ar(SinOsc.ar([
                    periodico.linexp(0.1**17, 1, 222, 6002),
                    complejo.linexp(0.1**17, 1, 309, 1009),
                    fijo.linexp(0.1**17, 1, 305, 380),
                    periodico.linexp(0.1**17, 1, 513, 6000),
                ], 0, 0.1), complejo.linexp(0.1**17, 0.9, -0.5, 0.5),
0.2));
            })
    });
```

```
56
57      //complejos
58
59      if((complejo > 0.9999),{
60          Ndef(\buf, {
61              var sig, env;
62              sig = PlayBuf.ar(2, ~complejo[0..~complejo.size].choose,
    (-7..-1).choose.midiratio, 1, doneAction: 2);
63              env = EnvGen.kr(Env.new([0, 0.6, 0.5, 0], [2, 25, 2]));
64              sig = sig * env;
65          })
66      });
67
68      if((complejo > 0.01) && (complejo < 0.5),{
69          Ndef(\buf, {
70              var sig, env;
71              sig = PlayBuf.ar(2, ~fijo[0..~fijo.size].choose, (-8..-1).
    choose.midiratio, 1, doneAction: 2);
72              env = EnvGen.kr(Env.new([0, 0.3, 0.3, 0], [2, 25, 2]));
73              sig = sig * env;
74          })
75      });
76
77      //Fijos
78
79      if((fijo > 0.3) && (fijo < 0.6),{
80          Synth(\caotico, [\buf, ~caotico[0..~caotico.size].choose, \
    rate, (-5..5).choose.midiratio])
81      });
82
83      if((fijo > 0) && (fijo < 0.9),{
84          Ndef(\sin,
85              {
86                  |norm = 0.2, dryrevlev= 0.05, controlfreq, room = 20,
    newfreq|
87                  var in, freq, hasfreq, fft, amp, loudness, noise,
    brown_noise;
88                  var in_Gen, rev, env, onset, sig, ctrspec;
89
```

```
                in = SoundIn.ar(0);
                amp = Amplitude.kr(in, 0.01, 0.01);

                #freq, hasfreq = Tartini.kr(in, 20);
                fft = FFT(h, in);
                loudness = Loudness.kr(fft);
                onset = Onsets.kr(fft, 0.01);
                newfreq = freq.lag(10);

                brown_noise = WhiteNoise.ar([0.5,0.5], amp*0.8);
                env = EnvGen.kr(Env.new([0, 0.7, 0.5, 0 ], [0.5, 0.5,
    1.1], 1, 2, nil), onset);
                noise = brown_noise * env;
                sig = Resonz.ar(noise, newfreq*0.99, 0.0001);
                //noise = HPF.ar(noise, 440);
                //Pan2.ar(in*noise, LFSaw.kr(loudness.linlin(20, 45,
    2, 0.01)), 1);
                rev = Mix.ar(GVerb.ar(sig, room, 5, 0.8, 0.8, 10, 1));
                Normalizer.ar(DelayN.ar(rev, 10, 2.5), norm, 0.01);
            })
        });

        //Periodicos

        if((periodico > 0.8) && (periodico < 1),{
            Synth(\periodico, [\buf, ~periodico[1..~periodico.size].choose
    , \rate, (-6..-5).choose.midiratio])
        });

},'/clase',recvPort: 5006);
```

Segunda aproximación

En esta aproximación los nuevos ejemplos de audio que son entregados al modelo generado, y segmentados cada dos segundos o alguna ventana temporal fija. Mediante essentia, se extraen las características de estos segmentos y los datos son enviados posteriormente a TensorFlow Server para entregar una nueva predicción. Una vez que la información detectada sobre el segmento de audio ha sido clasificada, ésta se envía

a Python vía un mensaje en formato JSON y Python se encarga de convertirla a OSC para que SuperCollider la reciba. A continuación los códigos correspondientes.

3.6.1. Segmentación y extracción de características en tiempo–real con Essentia

```
1
2  sampleRate = 44100
3  frameSize = 2048
4  hopSize = 2048
5  numberBands = 3
6  loudness = 1
7  patchSize = 20
8  displaySize = 10
9
10 bufferSize = patchSize * hopSize
11 buffer = np.zeros(bufferSize, dtype='float32')
12 vectorInput = VectorInput(buffer)
13 frameCutter = FrameCutter(frameSize=frameSize, hopSize=hopSize)
14 w = Windowing(type = 'hann')
15 spec = Spectrum()
16 mfcc = MFCC(numberCoefficients=13)
17 fft = FFT()
18 c2p = CartesianToPolar()
19
20 pool = Pool()
21
22 vectorInput.data  >> eqloud.signal >> frameCutter.signal
23 frameCutter.frame >> w.frame >> spec.frame
24 spec.spectrum     >> mfcc.spectrum
25 mfcc.bands        >> None
26 mfcc.mfcc         >> (pool, 'mfcc')
27 w.frame           >> fft.frame
28 fft.fft           >> c2p.complex
29 c2p.magnitude     >> onset.spectrum
30 c2p.phase         >> onset.phase
31 onset.onsetDetection >> (pool, 'onset')
32
33 def callback(data):
34     buffer[:] = array(unpack('f' * bufferSize, data))
```

```
35      mfccBuffer = np.zeros([numberBands])
36      reset(vectorInput)
37      run(vectorInput)
38      mfccBuffer = np.roll(mfccBuffer, -patchSize)
39      mfccBuffer = pool['mfcc'][-patchSize]
40      features = mfccBuffer
41      features = features.tolist()
42      return features
43
44  def tf_handler(args):
45    headers = {"content-type": "application/json"}
46    data = {"instances": [args]}
47    r = requests.post(url = "http://localhost:8501/v1/models/
        improv_class:predict", data=json.dumps(data), headers=headers)
48    response = r.json()
49    data = response["predictions"]
50
51    clases=data[0]
52    event = max(clases)
53    index = clases.index(event)
54
55  with sc.all_microphones(include_loopback=True)[-1].recorder(samplerate
      =sampleRate) as mic:
56    while True:
57      tf_handler(callback(mic.record(numframes=bufferSize).mean(axis=1))
       )
```

Derivado de todos los datos sobre la predicción que devuelve el modelo, existen varias aproximaciones para trabajar con esta información y pensar en cuáles podrían ser las las cualidades interactivas de SEALI. Una posibilidad sería hacer la sonificación de cada una de las clases que arroja el modelo el cual varía su índice de certeza dependiendo lo que haya detectado. Los valores de esta lista podrían ser asignados a ciertos parámetros para modificar algún sintetizador de acuerdo a la detección sonora de ese momento. Esta aproximación podría llamarse continua debido a que todos los valores de la predicción son tomados en cuenta y hay una continuidad numérica en los niveles de detección por cada clase. Por ejemplo si tuviéramos un modelo que fue entrenado con 10 clases, éste regresará 10 valores diferentes que en total suman uno,

y, en este caso, el número más alto se toma como la clase predominante en la predicción. La segunda sería una versión discontinua con propiedades deterministas. Ya que toma como válida la detección que predomina de un momento sonoro específico, de aquí se podría asignar la activación específica de un proceso ligado a la detección de alguna de las clases mediante reglas del booleanas. Cabe señalar que estas dos posibilidades continua/discontinua podrían combinarse para enriquecer las posibilidades interactivas de SEALI, es decir, cuando detecte una clase en específico, pueda activar un sintetizador o algún otro proceso, donde sus parámetros sean modulados continuamente por los valores que envía el modelo de predicción.

Con esta amplia metodología que integra varios algoritmos ejecutándose en tiempo–real que incluyen: la activación de una entrada de audio, la segmentación de audio entrante, la extracción de características del audio, la transformación de los datos analizados para su futura predicción y el envío al módulo de generación sonora para la interactividad; dejando de lado el hecho de que tiene que escuchar antes de tomar una acción, un número determinado de cuadros para poder interactuar en términos de coherencia tímbrica, es posible apreciar que SEALI reacciona de manera casi inmediata a lo que escucha. Su forma de reacción hasta este momento en el desarrollo, estaría ligada al estímulo-respuesta, la impulsividad, el instinto o la inmediatez humana. Finalmente este instinto está modelado por mi, SEALI, por ahora, no tiene ninguna agencia sobre la toma de decisión en este nivel interactivo, más allá de la forma en la que clasificó los materiales y la detección que hace en un momento determinado sumado a la contingencia de varios factores, ya que su predicción puede estar influenciada por ruidos externos, sonidos producidos por los improvisadores-intérpretes-músicos y los sonidos producidos por SEALI, que indudablemente afectan el resultado de la predicción. Una forma de atenuar (hasta el momento aún no implementada)esta situación sería entrenar al sistema con ejemplos de este tipo de texturas sonoras mixtas, ya que le podrían dar alguna pauta a SEALI del propio proceso de retroalimentación al que está sujeto, producto de todas las interacciones producidas por los agentes del sistema, y, tomar una acción diferente partiendo de tal detección.

A continuación presento una improvisación producida de la interacción de SEALI con la misma salida de audio que devuelve supercollider, es decir, escuchando sus propios procesos de retroalimentación:

`https://archive.org/details/seali-sola-23-nov-2020`

También presentó una improvisación que realicé junto a SEALI para el coloquio de alumnos del posgrado en 2021. En ella exploré mediante la guitarra eléctrica, algunos objetos (varilla, piñon, silbato de de la muerte), mi respiración y mi voz, ciertas sonoridades que me permitieron dialogar con SEALI a través de la activación de algunas de las clases con las que el modelo fue entrenado para esta versión. Además, para este performance realicé los visuales que acompañan la improvisación con el entorno de programación visual Hydra, así como algunas modificaciones, "al vuelo", a SEALI que me permitieron modular algunos de los parámetros de los sintetizadores del módulo de generación sonora. A continuación el enlace a esta performance:

`https://www.youtube.com/watch?v=ERfJlKyQ_wo&ab_channel=AaronEscobar`

Definitivamente hay muchos aspectos que mejorar en términos técnicos para poder contar con una integración más orgánica de los elementos sonoros que SEALI propone y los gestos sonoros que le son entregados a SEALI. Si bien como podemos escuchar en estas improvisaciones es posible crear una forma musical, ésta emerge de manera espontánea y sin un conocimiento explícito sobre lo que implica abordar este elemento. En la sección *Memoria a corto y largo plazo* del siguiente capítulo voy a indagar sobre este concepto y cómo SEALI se aproxima al análisis de la forma de la libre improvisación. Desde esta perspectiva, SEALI podría tener mayor incidencia en cómo lanza los materiales y cómo regula a gran escala las densidades sonoras, la tímbrica y la amplitud, más allá de solo reaccionar inmediatamente de manera impulsiva a lo que escucha.

Capítulo 4

SEALI: Escuchar la estructura en la música libremente improvisada

Antecedentes

Como primer antecedente quisiera destacar la problemática que el Dr. Hugo Solís destaca en su tesis *Understanding Collective Gestural Improvisations; a Computational Approach* sobre la imposibilidad de detectar la estructura musical limitándonos solo al análisis descriptivo de una señal de audio y la necesidad que esto implica para establecer marcos tecnológicos y sociales que permitan aproximarnos al análisis de la forma en la improvisación libre:

> Las herramientas que ofrece el procesamiento de señales no son suficientes para tener una idea básica de la estructura, patrones y comportamientos derivados de la señal. Así, otros campos deberían apoyar nuestra investigación: Machine Learning y Data Mining. Estos dos campos de la informática son dos campos relativamente nuevos que integran una amplia colección de metodologías, técnicas y algoritmos que ayudan a comprender mejor los datos sin procesar. [...] [Derivando] conclusiones significativas de las improvisaciones colectivas [...]. [...] La música no es sólo una señal. La

> música tiene un componente social y cultural que incluso puede anular cualquier tipo de conclusión derivada únicamente del análisis de los datos acústicos. Por ello, si pretendemos estudiar la interacción entre músicos mientras realizan improvisaciones libres, es importante implementar no solo el análisis de audio sino también utilizar algunas de las técnicas de modelado de redes sociales e interacción humana.(Solís, 2006)

Dado este panorama, en un momento inicial, me propuse dotar a SEALI de algunas características que le permitieran abordar estructuras sintagmáticas de nivel superior (frases, motivos, períodos, secciones) y así poder incidir en términos de amplitud, gestualidad y oleadas energéticas en el desarrollo de una improvisación libre. Ello fue posible a través de crear un modelo computacional alimentado por distintas curvas paramétricas (las cuales tomaron la evolución de la amplitud y diversos aspectos tímbricos, de un conjunto de improvisaciones, a varios niveles de temporalidad –pasos de tiempo–) que responden al nivel de la estructura musical y que dieran cuenta de la evolución temporal de las improvisaciones con las que fue entrenado el modelo.

Como trabajo previo, para abordar esta aproximación estudié las cadenas de Markov. Ellas me permitieron generar un modelo predictivo de la forma el cual toma en cuenta un número finito de estados previos para predecir los posteriores de acuerdo con un corpus de estados. El proyecto que generé a este respecto, dado el estado en el que nos encontrábamos en ese momento al inicio de la pandemia en 2020, fue Markovirus-19.[1] Este sistema toma como estados posteriores posibles las palabras con las que fue dotado. En contraste, para predecir la estructura de una improvisación, las cadenas de Markov tomarían los grados de diferenciación del audio como clases posibles a las cuales referirse, dado un número finito de pasos anteriores para predecir la siguiente clase. De ahí tomé los descriptores que definen esas diferencias y medí la distancia entre ellas, entonces, en vez de comparar estados de palabras (como sucede en Markovirus-19) SEALI compara entre las diferentes secuencias de clases basadas en la homogeneidad tímbrica, la amplitud y el contraste. El procedimiento

[1] Generador automático de contenidos sobre las últimas noticias y reflexiones en torno al Covid-19. `https://replit.com/@Atsintli/Markovid-19#main.py`

que seguí para esto fue muy similar al descrito en el capítulo 3: primero se segmenta el audio siguiendo una escala temporal más amplia (p. Ej. 6, 18, 36 segundos), se extraen las características de audio, se envían al modelo generado previamente para obtener la clase a la que pertenece cada segmento. Estos datos se escriben en un archivo *csv* con la secuencia de predicciones de toda una serie de archivos de audio, creando así una base de datos anotada por el propio modelo de predicción. El archivo csv generado es analizado en tiempo–real por una cadena de Markov. Después, un algoritmo compara la secuencia sonora actual con la base de datos de la evolución de clases en el tiempo. Busca si el fragmento se parece a algún otro dentro de su base de datos, dependiendo de este resultado el sistema toma la decisión de seleccionar un nuevo segmento de acuerdo con la estructura analizada. Posteriormente, activa una acción sonora congruente con el desarrollo arquetípico de las improvisaciones contenidas en su base de datos. Estas acciones podrían estar condicionadas y afectar elementos musicales de alto nivel, como la agógica, el desarrollo tímbrico general, la amplitud, el índice de cambio, la homogeneidad, el contraste, la novedad, etc.

Si bien las cadenas de Markov pueden hacer análisis multivariable y se pueden implementar cadenas de orden variable, es decir que a través de ciertas reglas, cambien en el tiempo el número de pasos previos para predecir la siguiente secuencia, no seguí explorando esta aproximación dado que en ese momento fue de mi interés seguir abordando perspectivas de análisis de series de tiempo vinculadas con el aprendizaje de máquina y las redes neuronales recurrentes.

Estado actual

Más allá de la respuesta inmediata de SEALI a lo que escucha, una parte importante apunta al análisis de la forma o de estructuras sintagmáticas de nivel superior. A partir de éstas, SEALI podría tomar decisiones a un nivel superior que den cuenta de las oleadas energéticas de un contexto más amplio de la libre improvisación. Este conocimiento sobre lo que ocurre, por ejemplo a nivel de amplitud, le podría dar una certeza sobre la evolución en términos de volumen y, de ahí, tomar partido en el desarrollo temporal de la intensidad dentro de un contexto sonoro. Esto también se podría extrapolar a la capacidad del sistema de saber en qué momento de la impro-

visación se encuentra, ya sea el inicio, el intermedio o el final. La propuesta general de este capítulo es abordar la creación de modelos computacionales que contengan la información sobre lo que ocurre en la estructura de la libre improvisación a distintos niveles temporales. Los elementos a considerar en la construcción de estos modelos son: evolución en amplitud, índice de cambio espectral, timbre y aplanamiento espectral. También podríamos agregar alguna descripción de alto nivel como humor musical (energético, lento, aburrido, triste, alegre, oscuro, brillante) o establecer jerarquías temporales basadas en la homogeneidad tímbrica, la repetición, el contraste o la novedad, si bien estas características no son exploradas en esta investigación, podrían resultar muy interesantes para el análisis de la estructura.

Todo este despliegue en un inicio conceptual, me llevó a dedicar un tiempo considerable para familiarizarme con las redes neuronales recurrentes para la predicción de secuencias temporales en el contexto de la improvisación libre. Estas secuencias temporales, analizadas a diferentes escalas, pueden ser de utilidad para definir en términos numéricos la estructura de una improvisación. Además, nos puede ayudar a determinar cómo es su cualidad armónica o inarmónica (*spectral flatness*, su espectro energético (*loudness*), cuál es su estructura en cuanto a centro espectral (*spectral centroid*) o su timbre general (*MFCCs*). Para ejemplificar estás características del análisis del audio, muestro en las siguientes figuras el análisis generado por los descriptores de audio de la biblioteca de audio *Essentia*[2], de una improvisación libre *open paper tree* creada por Michel Doneda, Paul Rogers y Lê Quan Ninh.

Partir desde esta perspectiva, me ha llevado a explorar una técnica del aprendizaje de máquina que se ha vuelto muy popular dadas sus capacidades para predecir, a distintos niveles temporales, estados subsecuentes de un conjunto de datos, dada una secuencia inicial de valores. El nombre de este algoritmo es *Long Short Term Memory* LSMT por sus siglas en inglés.

Las redes LSTM han demostrado tener buen desempeño en la resolución de este tipo de problemas y ser excelentes para predecir nuevos datos a partir de una secuencia inicial. Para demostrarlo, realicé un ejercicio preliminar con las variaciones *Golberg* de J.S. Bach. Este ejercicio lo realicé debido a que mi asesor principal el Dr. Hugo Solís,

[2] `essentia.upf.edu/`

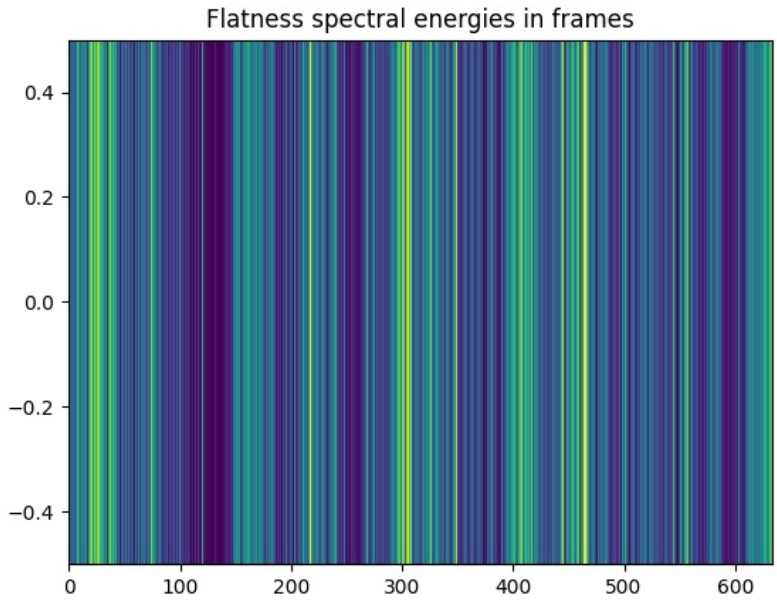

Figura 4.1: La planitud espectral o coeficiente de tonalidad, se utiliza para caracterizar un espectro de audio y describe una forma de cuantificar qué tanto tono o ruido tiene un sonido.

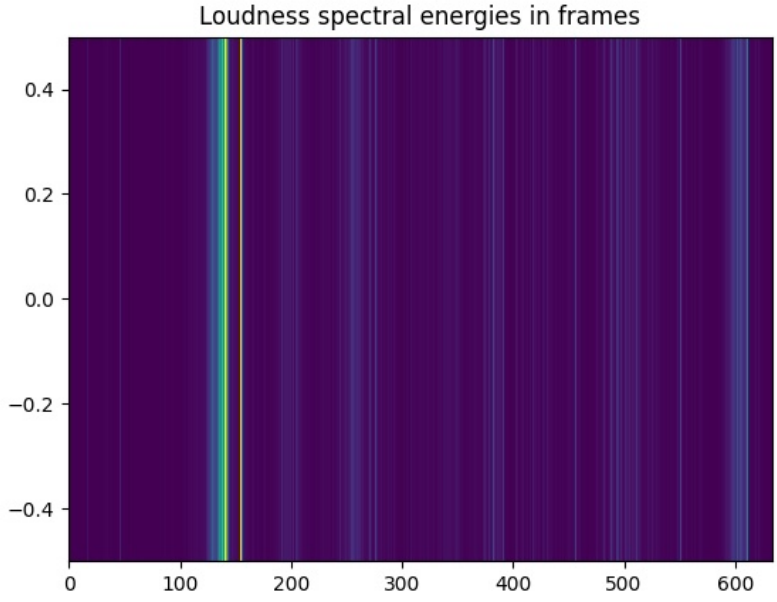

Figura 4.2: La fuerza, intensidad o volumen espectral, es la cantidad de presión sonora desde una perspectiva psicoacústica.

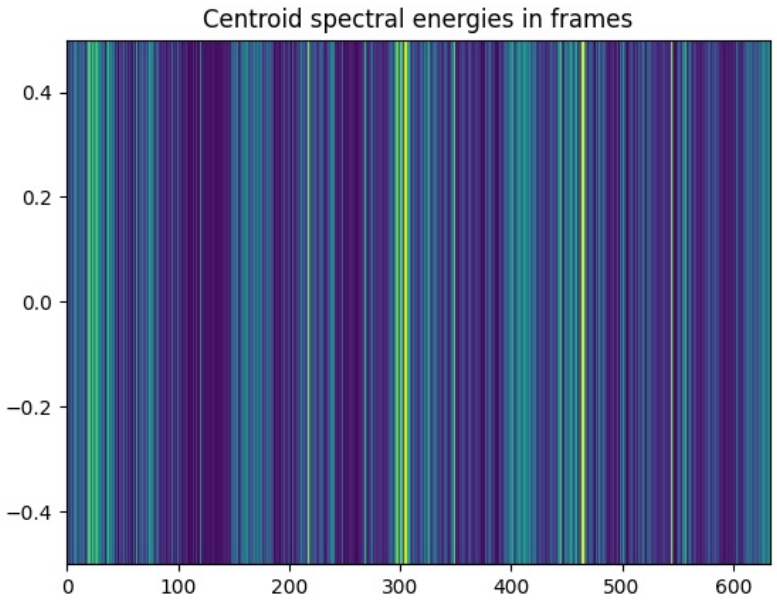

Figura 4.3: El centroide espectral es la medida empleada para caracterizar y encontrar la masa del espectro de una señal de audio. Perceptiblemente, tiene una conexión directa con el brillo de un sonido.

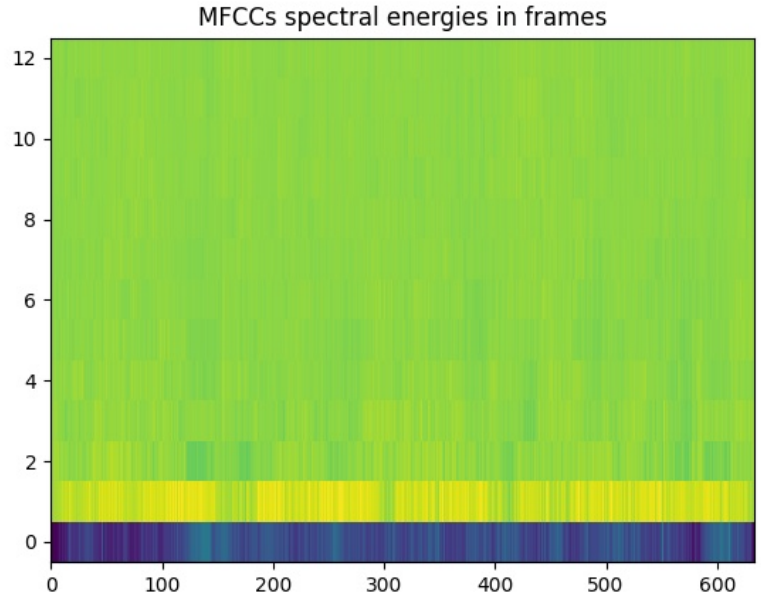

Figura 4.4: Los MFCCs se utilizan para describir el espectro del audio tomando una transformada de coseno inversa de un espectro de potencia logarítmica en la escala de mel (adaptación psicoacústica no lineal sino exponencial). Preceptivamente, tiene una conexión con el timbre sonoro.

sugirió realizar algunos ejercicios que demuestren la fiabilidad de esta aproximación al emplearla a problemas musicales reconocibles fácilmente por una escucha musical occidental ligada a la tonalidad el ritmo y la amplitud. El objetivo de este ejercicio fue diagnosticar a grandes rasgos las posibilidades y limitantes de está herramienta, para así, determinar algunas de las variables de entrada y de estado necesarias para generar reconstrucciones congruentes a nivel armónico dentro del lenguaje musical de Bach. Partir de este ejercicio, me fue de utilidad para demostrar que las redes LSTM pueden generar un tipo de aproximación a la forma musical; ya que, dada una secuencia inicial pueden reconstruir los elementos subsecuentes partiendo de la base de datos de entrenamiento. Asimismo, al realizar este experimento, son reconocibles las secciones musicales del tipo AABB presentes en las variaciones Goldberg. Esto me ha ayudado a entender cómo funcionan las redes LSTM, cómo se desempeñan mejor, qué tipo de transformaciones es necesario hacer a los datos de entrada y evaluar su desempeño en contextos reconocibles para mi memoria musical en un contexto tonal.[3][4]

Una característica que detecté al trabajar con este ejemplo fue darme cuenta que hay un umbral muy interesante a explorar en las redes LSTM, donde un tipo de "creatividad computacional" emerge momentos antes del estado de convergencia del modelo. Por un lado, si el modelo está completamente entrenado –ha convergido–[5], al entregarle una secuencia original (incluida en los datos de entrenamiento de la red) y hacer predicciones de manera iterativa, el modelo llega a generar una recreación exactamente fiel a la obra original de Bach. Por otro lado, si el modelo no ha convergido, en etapas iniciales del entrenamiento, genera una predicción casi nula sobre la serie de datos entregada; generando, mayoritariamente notas al unísono. Encontrar la época[6] adecuada entre estos dos momentos del entrenamiento, podría generar una reconstrucción de la secuencia mucho más abierta que, en algunos casos, podría presentar ciertos rasgos de creatividad computacional inédita. Esta época se encuentra en un estado donde el entrenamiento está ligeramente por encima de la mitad del

[3]El código para realizar estos experimentos fue derivado del siguiente desarrollo de Sigurður Skúli Sigurgeirsson Skuldur.`https://github.com/Skuldur/Classical-Piano-Composer`

[4]Los archivos MIDI para el análisis se obtutieron del siguiente enlace.`https://www.classicalarchives.com/newca/`

[5]El punto donde el modelo ya no puede aprender más de los datos presentados.

[6]La época es el número de iteraciones por las que pasa el modelo al ser entrenado.

proceso completo de entrenamiento y antes del punto de convergencia del modelo. Más concretamente, en estos experimentos, este umbral de creatividad lo encontré oscilando entre un 65 y 85 por ciento de certeza en el proceso de aprendizaje del modelo. Para cerrar con este experimento, en el siguiente enlace comparto algunos de los ejemplos generados con versiones del modelo en estado de entrenamiento de 65, 70 y 95 por ciento de certeza en su entrenamiento.[7]

A través de realizar éstos experimentos, concluí que al integrar las redes LSTM a SEALI, sería capaz de predecir hacia dónde podría conducirse una improvisación de acuerdo con una base de datos de improvisaciones compuesta por distintas características del contenido espectral en la señal de audio. Esta aproximación tiene la finalidad de hacer predicciones sobre elementos de las improvisaciones con las que fue entrenado un modelo, así como elementos nunca antes vistos en contextos improvisatorios de manera coherente. De este modo, los objetivos de este capítulo son:

1. A partir de una secuencia inicial dada, comprobar si las redes LSTM son capaces de reconstruir en tiempo diferido, utilizando segmentos de audio, improvisaciones libres que puedan reflejar la forma musical presente en esta práctica.[8]

2. Aplicar este conocimiento a una implementación en tiempo–real que permita ejecutar un autómata musical capaz de interactuar de forma coherente en la improvisación libre tras haber escuchado algunos segundos del audio de entrada. Para ello, primero realizaría predicciones de dónde podría dirigirse la improvisación, y, después, activaría fragmentos sonoros similares o contrastantes, realizando procesamientos a la señal de audio o modulando la amplitud general, de manera que éstas acciones correspondan con lo que podría ocurrir en una improvisación real.

[7] `https://archive.org/details/goldberg-65/`

[8] Desde esta aproximación, sería interesante observar qué tipo de resultados se generan al variar el tamaño de las secuencias, la cantidad de descriptores a considerar para entrenar al modelo o el número de parámetros de la arquitectura de la red LSTM.

4.1. Predicción de series de tiempo: Memoria a corto y largo plazo

Para comenzar el abordaje de las redes LSTM, considero de suma importancia definirlas, saber para qué son útiles, así como describir a grandes rasgos su funcionamiento interno.

Las redes LSTM (Long-Short Term Memory) son redes neuronales recurrentes (RNN) que pertenecen al campo denominado redes neuronales profundas (Deep Neural Networks) o aprendizaje profundo. Las redes recurrentes artificiales son capaces de recibir una secuencia de datos de entrada X de la cual almacena un estado oculto anterior y un estado actual, para entregar una predicción (Y) y el valor actualizado del estado oculto. Estas predicciones son calculadas partiendo de los estados ocultos anteriores, esto quiere decir que las predicciones son dependientes de los estados anteriores de las mismas predicciones. Así es posible observar la predicción de un momento "x" en relación con los estados ocultos anteriores. Esto funciona al pasar los estados ocultos por tres funciones anidadas de tangentes hiperbólicas dentro de la función de activación *Softmax* la cual generalmente regresa una predicción multivariable. Una de las desventajas de las RNN es su memoria a corto plazo ya que al activar las funciones tangentes de forma anidada sobre las predicciones anteriores, los resultados de estas operaciones afectan muy poco al trabajar con secuencias muy largas. Mientras más largas las secuencias, el valor de la predicción se acerca más a cero, nulificando el efecto que podrían tener los elementos anteriores sobre elementos subsecuentes. Este comportamiento obliga a que las secuencias procesadas sean muy cortas para que el efecto de los estados ocultos tengan una injerencia sobre las predicciones subsecuentes.

Las redes LSTM fueron diseñadas específicamente para lidiar con este problema y posibilitar la predicción certera de secuencias mucho más largas. Estas redes son capaces de recordar y preservar datos relevantes en las secuencias de tiempo a largo y corto plazo. Sus características le permiten priorizar sobre datos considerados como relevantes para el procesamiento de las secuencias de entrada. Una red LSTM tiene las mismas características de entrada y salida que la RNN salvo que se agrega un

tercer elemento a su arquitectura; una *celda de estado*, en inglés, *cell state*. Esta funciona como un espacio de almacenaje al cual se agregan datos relevantes (los otros se olvidan), permitiendo manipular la memoria de la red. La celda de estado tiene tres compuertas: *forget gate*, la cual elimina elementos irrelevantes de la memoria, *update gate*, añade elementos relevantes a la memoria y *output gate*, crea el estado oculto actualizado. A su vez, estas compuertas tienen tres elementos: una red neuronal, una función de activación sigmoidal y un multiplicador. Analicemos que pasa internamente en estas tres compuertas:

El proceso comienza con la *forget gate* la cual toma el estado oculto anterior y el estado actual y los pasa por una función de activación sigmoidal. Si los valores arrojados por la función de activación son cercanos a 0 la LSTM descartará esa información, considerándola como no relevante. Por el contrario, si los valores arrojados tras la función de activación son cercanos a uno, la información se conservará en la celda de estado. Generando así, un primer vector de salida *Ft*. En la compuerta de actualización, *update gate*, el estado oculto anterior y la entrada actual pasan por la función de activación sigmoidal, aquí, al igual que en la compuerta anterior, se aprenden pesos y sesgos durante el entrenamiento de la red. Posteriormente la compuerta genera un nuevo vector de salida *Lt*, que conserva los valores cercanos a uno y elimina la información cercana a cero. La celda de estado es actualizada al multiplicar la salida de esta celda (*update gate*) por el vector generado en la compuerta anterior (*forget gate*). Después se crea un nuevo vector *Ot* de valores candidatos de la nueva memoria, donde también se aprenden los pesos y sesgos de este proceso. Nuevamente se filtran los valores al multiplicar punto a punto el vector obtenido por el generado en la compuerta *update* y el resultado se suma a los valores anteriores de la celda de estado (generados por la *forget gate*) generando así la memoria actualizada. Posteriormente, en la *output gate* se calcula el nuevo estado oculto, este estado de salida es una versión filtrada del estado de celda *update gate*. En esta compuerta la celda de estado es escalada al rango de -1 y 1 (el rango que tiene el estado oculto) con la función de tangente hiperbólica. Finalmente se filtran los valores al multiplicar punto a punto el vector escalado por el vector generado por la compuerta de salida.

Determinando así las porciones de la celda de estado que formarán parte del nuevo vector *at*.

Otro elemento importante a considerar son los "pasos de tiempo" o *timesteps*, los cuales almacenan el tamaño de la secuencia a aprender por la red LSTM. Cabe mencionar que a cada paso de tiempo le corresponde una red LSTM, correspondiendo a cada uno de los instantes de tiempo de la secuencia.

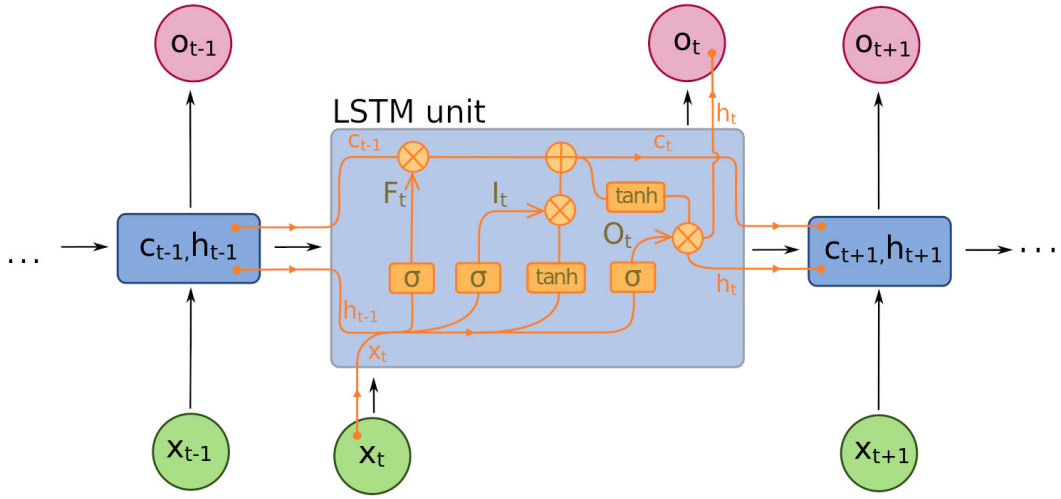

Figura 4.5: Arquitectura de la unidad LSTM

Las aplicaciones que podemos encontrar para las LSTMs son muy variadas y van desde predicciones médicas, bursátiles, de mercado, pasando por el pronóstico de fenómenos medioambientales, el manejo de aeropuertos, diseño de fármacos, hasta la generación de diálogos, texto, traducciones, video y la composición musical.

Una desventaja considerable de este tipo de redes neuronales es que requieren de mucho poder de cómputo para poder entrenar modelos y llevarlos a converger, es decir, el momento del entrenamiento donde el modelo alcanza un grado de optimización en el cual, la pérdida *loss* (que tan lejos está el modelo de llegar al punto de optimización de aprendizaje) y la validación *validation* (proceso para validar la calidad del modelo mediante el conjunto de datos para pruebas) cambian poco o nada después de cierto número de iteraciones. Ejemplo de ello sería llegar al punto del entrenamiento donde el modelo puede predecir, dada una serie de valores de entrada, certeramente todos los casos. Si los datos de entrada para verificar el fun-

cionamiento del modelo fueran los mismos con los que fue entrenado, y este proceso de predicciones fuera realizado iterativamente, tendría que entregar series idénticas a las originales.[9]), normalizar las funciones de activación en una capa oculta (*Batch Normalization*), si se usará algún otro método de regresión intermedio entre las capas de las celdas LSTM, ejemplo *RNNs* o *GRUs* (Gated Recurrent Units), el tamaño del lote de archivos a procesar por cada iteración *Batch Size*. Y finalmente, el tipo de activaciones, el optimizador y la pérdida.

Todos los parámetros anteriores junto con la base de datos, su tamaño, características y complejidad (en términos de variabilidad); juegan un papel determinante para la creación de un modelo eficiente. Todos estos elementos que conforman la materia de la arquitectura de la red neuronal funcionan de manera entrelazada, es decir, están en un constante flujo e intercambio de información, de ello depende la generación de un buen o mal entrenamiento para un modelo.

4.2. Recomposición automática en tiempo diferido con LSTM uno a varios

Vamos a desarrollar el objetivo número uno de este capítulo. Para comenzar, parto de la creación de varios modelos predictivos de la estructura que toman en cuenta los estados actuales para predecir los estados posteriores de acuerdo con la base de datos utilizada para entrenar al modelo. Abordaré a través del análisis de algunas improvisaciones tanto mías como de algunos improvisadores, los problemas, soluciones, derivas y aciertos que he explorado en torno al tema de la predicción de la forma en la improvisación libre.

More 74 - Derek Bailey

Para el primer experimento utilicé el disco del improvisador Derek Bailey *More74*, este disco contiene 13 improvisaciones libres realizadas con guitarra eléctrica. Para su

[9]Para mayor información sobre estos y otros términos relevantes al aprendizaje de máquina, sugiero consultar el siguiente glosario. [10] Para poder llegar a converger un modelo y predecir certeramente datos que el modelo no contiene, es necesario ajustar una serie de parámetros de la arquitectura de la red. Estos pueden ir desde el número de capas LSTM, número de neuronas por capa, decidir si se usará algún tipo de regularización en los datos procesados (*Dropout*[11] del modelo, esto imposibilita su capacidad de hacer predicciones sobre ejemplos nunca antes vistos.

análisis, primero segmenté todos los audios con base en cambios tímbricos mediante el algoritmo *SBIC* y MFCCs, ambos de la librería *essentia*, resultado un total de 17802 segmentos de audio que van de 100 milisegundos a 1600 milisegundos (o 1.60 segundos).

Después extraje sus características con el descriptor de audio *Flatness* para medir el nivel máximo de ruido o sutileza tímbrica de cada segmento, estas características de audio fueron guardadas en un archivo csv. Posterior a esto, generé un conversor de csv a json donde redondeé todos los datos a un entero y dos decimales lo que da un máximo de 101 (del 0 al 100) posibles elementos dentro de toda la base de datos. Esto lo realicé con el objetivo de reducir el nivel de complejidad resultante del análisis y tener mayor recurrencia en la base de datos, ya que cualquiera de los descriptores de audio de la librería *Essentia* generalmente tienen de 11 a 20 decimales. Entrenar al modelo con esta cantidad de decimales por cada elemento, llegaría a ser una tarea casi imposible de resolver, dado que la recurrencia en los datos podría incluso no existir; se requeriría una máquina muy potente y una base de datos enorme para poder encontrar recurrencias con una resolución de datos tan grande. De ahí surge la importancia de la reducción de la complejidad de los datos. Esta transformación de las de muestras, también contempla su reorganización en elementos individuales y posteriormente se les asigna una clase de acuerdo con en el número al que corresponde dada la gradación de 101 elementos, dando como resultado un ordenamiento secuencial de los 17802 segmentos con la siguiente estructura:


```
{
 "clase": 74,
 "features": 0.79,
 "file": "000000"
},
{
 "clase": 70,
 "features": 0.73,
 "file": "000001"
},
```

```
{
 "clase": 9,
 "features": 0.1,
 "file": "000002"
},
{
 "clase": 8,
 "features": 0.09,
 "file": "000003"
}
```


Cabe destacar que de la gradación de 101 elementos es muy probable que en el lenguaje musical de un solo improvisador, ciertos elementos no estén presentes, en el caso del álbum de Bailey solamente habría 79 clases que corresponden al aplanamiento espectral (spectral flatness).

Posteriormente estos datos fueron procesados por un nuevo algoritmo que generó secuencias basadas en pasos de tiempo (*timesteps*). Para cada caso, estas secuencias fueron agrupadas en n elementos (tomando como punto de partida n=100), a los cuales les corresponde un siguiente elemento convertido en *dummy variables* o *one hot encode*. Esto significa que cada clase será representada un grupo de varios ceros y un solo uno de acuerdo con el número de la clase. Por ejemplo si tuviéramos 5 clases la conversión se vería de la siguiente manera:

```
[0,0,0,0,0]
[0,1,0,0,0]
[0,0,1,0,0]
[0,0,0,1,0]
[0,0,0,0,1]
```

Adicionalmente cada segmento de paso de tiempo se recorre de uno en uno, de tal manera que la estructuración de los 17802 elementos secuenciados en 100 pasos de tiempo, generaron un total de 17802 listas de 100 elementos a analizar secuencialmente. Suponiendo que tenemos 10 posibles clases, en el siguiente ejemplo

podemos observar la estructuración de 4 secuencias genéricas basadas en 5 pasos de tiempo y su clase correspondiente:

```
timeSteps = 5

x = [
            [[0],[1],[2],[3],[4]],
    [[1],[2],[3],[4],[5]],
    [[2],[3],[4],[5],[6]],
    [[3],[4],[5],[6],[7]],
    ]

y = [4,5,6,7]

Conversión de ‘‘y’’ a \emph{dummy variables} o \emph{one hot encode}:

y = [
    [0,0,0,1,0,0,0,0,0,0]
    [0,0,0,0,1,0,0,0,0,0]
    [0,0,0,0,0,1,0,0,0,0]
    [0,0,0,0,0,0,1,0,0,0]
    ]
```

Una vez almacenados los datos en estas tres variables *x, y, timeSteps* es posible asignarlas directamente a la red LSTM y comenzar con el proceso de entrenamiento.

Al entrenar un total de 1723 veces la red LSTM con esta base de datos, la gradiente de descenso o pérdida (loss) bajó hasta alcanzar un mínimo de 0.0100. Posterior a esto realicé algunas pruebas para comprobar si el modelo generado durante el entrenamiento era capaz de reconstruir secuencias iguales a las originales. Para ello, le pasé directamente la secuencia original que va del elemento 0 al 99 y el resultado fue que el modelo regresa el elemento 100 como siguiente valor dada esa secuencia numé-

rica inicial. Al generar iterativamente este proceso un total de 100 veces, donde las nuevas predicciones se convierten en nuevas entradas para las secuencias, fue posible detectar que el modelo es capaz de reconstruir perfectamente las secuencias derivadas de las secuencias originales, de manera tal que los elementos subsecuentes de las predicciones son tomados como nuevas entradas, lo cual posibilita una reconstrucción fiel de la secuencia original de la improvisación.

El objetivo de entregar los datos en el formato *dummy variables* o *one hot encode* al modelo, es poder hacer predicciones porcentuales de cada una de las clases, donde el total de la predicción suma 1 y la predicción se distribuye en el número total de clases. Si fueron 5 clases habría una distribución porcentual en cada una dependiendo la predicción del modelo.

Así, esta implementación hace una predicción basada en porcentajes de las 100 clases posibles, tomando como válida la clase con mayor porcentaje detectado. Esto es lo que se conoce como uno a muchos ya que al darle un solo vector como entrada, en este caso la descripción numérica del audio en términos de *flatness*, el modelo regresa una predicción basada en la distribución porcentual de las 79 clases detectadas en la base de datos.

```
[ 7.44522412e-12 2.04972995e-14 9.99999046e-01 4.51629484e-10
  7.66142982e-11 1.27537586e-12 9.34487616e-07 5.19589986e-19
  3.51618199e-17 5.40090314e-19 2.02918503e-20 1.35503249e-13
  2.30138107e-17 8.34981587e-18 4.31436952e-15 6.35501157e-12
  1.55823490e-21 1.30473660e-16 1.48664131e-19 8.83408314e-17
  8.12379815e-23 2.48768947e-20 3.57433835e-22 2.58888080e-20
  1.40611011e-22 2.77724262e-19 5.14375802e-19 6.76992396e-15
  7.78100287e-21 2.57830672e-19 6.09728219e-19 7.78904547e-22
  1.16600995e-20 4.95691005e-24 1.09669455e-20 8.92442278e-16
  8.08464100e-20 4.68868471e-18 4.10288258e-13 1.54392444e-18
  4.35330069e-16 2.15798505e-16 1.07322927e-11 1.00106108e-13
  3.84681919e-16 3.90128734e-18 2.26377152e-20 8.24967739e-17
  1.20065464e-17 1.46967215e-15 6.02851945e-20 3.75001219e-19
```

```
6.10228798e-18 7.10534498e-18 4.61690407e-20 8.85105785e-23
7.89202683e-21 4.67292625e-22 1.86136235e-18 3.08901598e-22
0.00000000e+00 1.11283659e-25 5.05030392e-23 1.55931161e-28
1.38302273e-22 2.20104688e-35 3.34486145e-26 0.00000000e+00
0.00000000e+00 0.00000000e+00 6.84090819e-35 1.17220011e-35
2.19553853e-34 9.99330176e-34 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00]
```

En este ejemplo podemos ver que la clase 2 (contando 0,1,2) fue la que obtuvo mayor porcentaje en la predicción con 9.99999046e-01. Esta clase será posteriormente reconvertida a su definición numérica de *flatness* y comparada con la base de datos en formato json basada en el ordenamiento secuencial.

```
Clase predicha: 2
Clase a flatness: 0.03
```

Este ejercicio es de gran ayuda para evaluar qué tanto puede predecir el modelo las secuencias siguientes dada una secuencia semilla original. Además, es de utilidad para generar reconstrucciones sonoras basadas en las predicciones del modelo. Para ello, las predicciones hechas son almacenadas en una lista que es comparada con la base de datos original de la cual se escogen secuencialmente los elementos con menor diferencia entre la predicción y la base de datos original. Los candidatos seleccionados son identificados de acuerdo a su nombre y la lista resultante es concatenada para generar una recreación de la pieza original.

En este ejemplo, proporcioné al modelo los primeros 100 elementos de una de las improvisaciones de Bailey y podemos ver en el resultado que los elementos escogidos por el modelo son exactamente los mismos que se esperan.

```
Predicciones subsecuentes de 20 elementos de la base de datos
Clase:
[1, 5, 6, 5, 3, 5, 6, 5, 4, 7, 7, 5, 4, 14, 19, 14, 5, 6, 5, 3]
```

Flatness:
[0.02, 0.06, 0.07, 0.06, 0.04, 0.06, 0.07, 0.06, 0.05, 0.08, 0.08, 0.06, 0.05,
0.15, 0.2, 0.15, 0.06, 0.07, 0.06, 0.04]

Lista de los 20 elementos subsecuentes de la base de datos original
{
"clase": 1,
"features": 0.02,
"file": "000100"
},
{
"clase": 5,
"features": 0.06,
"file": "000101"
},
{
"clase": 6,
"features": 0.07,
"file": "000102"
},
{
"clase": 5,
"features": 0.06,
"file": "000103"
},
{
"clase": 3,
"features": 0.04,
"file": "000104"
},
{
"clase": 5,

```
 "features": 0.06,
 "file": "000105"
},
{
 "clase": 6,
 "features": 0.07,
 "file": "000106"
},
{
 "clase": 5,
 "features": 0.06,
 "file": "000107"
},
{
 "clase": 4,
 "features": 0.05,
 "file": "000108"
},
{
 "clase": 7,
 "features": 0.08,
 "file": "000109"
},
{
 "clase": 7,
 "features": 0.08,
 "file": "000110"
},
{
 "clase": 5,
 "features": 0.06,
 "file": "000111"
```

```
},
{
 "clase": 4,
 "features": 0.05,
 "file": "000112"
},
{
 "clase": 14,
 "features": 0.15,
 "file": "000113"
},
{
 "clase": 19,
 "features": 0.2,
 "file": "000114"
},
{
 "clase": 14,
 "features": 0.15,
 "file": "000115"
},
{
 "clase": 5,
 "features": 0.06,
 "file": "000116"
},
{
 "clase": 6,
 "features": 0.07,
 "file": "000117"
},
{
```

```
 "clase": 5,
 "features": 0.06,
 "file": "000118"
},
{
 "clase": 3,
 "features": 0.04,
 "file": "000119"
},
```


Un aspecto interesante a resaltar, es saber qué pasa con el modelo en el caso de realizar una predicción sobre datos que no estuvieran en la base de datos con la que fue entrenado. En este caso, el modelo realiza una predicción basada en grados de cercanía numérica entre los descriptores con los que se entrenó el modelo y el elemento a predecir; es decir, al recibir un elemento nunca antes visto en su base de datos, su predicción sería el elemento más cercano dentro de los elementos de la base de datos con relación a ese nuevo elemento detectado. De ahí toma los descriptores que definen esas diferencias y mide la distancia entre cada uno de ellos, comparando entre las diferentes secuencias de clases basadas en la homogeneidad tímbrica, la amplitud, el contraste, etc.

Esta aproximación me es útil para verificar dos cosas: qué tanto cambia o qué tanto permanece la similitud numérica en la propuesta generada por el modelo LSTM con relación a las secuencias originales y a qué parámetros responde este comportamiento. Similar al ejemplo de las variaciones Goldberg, si el modelo ha convergido al analizar las secuencias numéricas, en teoría, podría generar una propuesta sonora, en términos de *flatness*, basada en el estilo particular que Derek Bailey propuso en *More 74*. Si se partiera de una secuencia completamente distinta a las secuencias reconocidas por el modelo, podríamos hablar de una forma de transferencia de conocimiento del propio estilo y aproximaciones improvisatorias de Bailey hacia un contexto, momento y lugar completamente distintos; esto siempre recordando los sesgos numéricos implícitos en el análisis. Teniendo esto en cuenta, se podría decir que la secuencia resultante sería una propuesta numérica para improvisar ligada a la estructuración generada al

analizar los datos de las improvisaciones de Bailey. Esta propuesta podría replicarse a varios niveles y con distintos descriptores de audio para analizar la forma en la que el sistema reconstruye a diferentes escalas temporales una improvisación.

Sink into Return + Movil II

Otra de las pruebas que realicé a este respecto, fue unir la improvisación *Sink into Return* de la improvisadora Clare Cooper y la pieza abierta *Movil II* del compositor Manuel Enríquez tocada por el violinista Fabian Rangel. Para delimitar estos ejemplos, fue necesario señalar un claro final entre ambos, así que después de ser analizados por el descriptor de audio *spectral flatness*, agregué "0.00" como valor del descriptor final después del conjunto de segmentos de cada ejemplo. Cabe señalar que incluir esta pieza abierta de Manuel Enriquez como ejemplo de análisis evidencia que el desarrollo de SEALI no solo se limita a trabajar con improvisación libre, sino que puede extenderse a más prácticas musicales muy diversas que desbordan y complementan el mundo de la improvisación libre. Para estas pruebas el modelo fue entrenado con secuencias de 5 pasos de tiempo. Cabe señalar, que todos los modelos generados a partir de la iteración 14557, con una gradiente de descenso inferior a 0.001, son capaces de generar al cien por ciento una recreación de los ejemplos originales. Para ambos casos, la base de datos tiene un total de 2076 muestras de las cuales se detectaron 58 elementos diferentes. Es importante notar que el punto de convergencia al analizar individualmente ambas performances, ocurrió mucho antes (alrededor de la iteración 7000, casi la mitad de 14557) que en la versión con los ejemplos concatenados.

Cabe mencionar que los datos recogidos por el descriptor de audio *flatness* de una improvisación a otra son diferentes y se mueven en espectros armónicos distintos. Como puede apreciarse en la figura 4.7 en *Sink into Return*, del segmento 0 al 1000 aproximadamente, hay un rango de acción en términos de planitud espectral que va de la clase 0.05 a 0.45, mientras que en el *Movil II*, del segmento 1000 al 2000, hay un espectro de acción mucho más amplio que va de la clase 0.03 a la 0.58. Esto deja ver el ámbito armónico de ambos instrumentos y la exploración musical de cada improvisador dentro de su práctica y estilo específico.

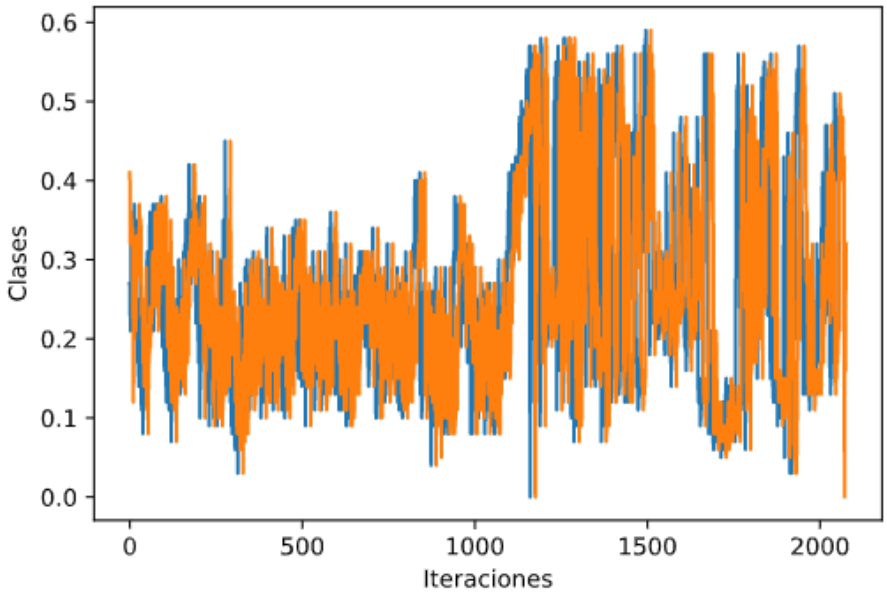

Figura 4.6: Predicción de toda la secuencia original (desfasada por 20 segmentos) basada en seleccionar los elementos iniciales que van del 1 al 5.

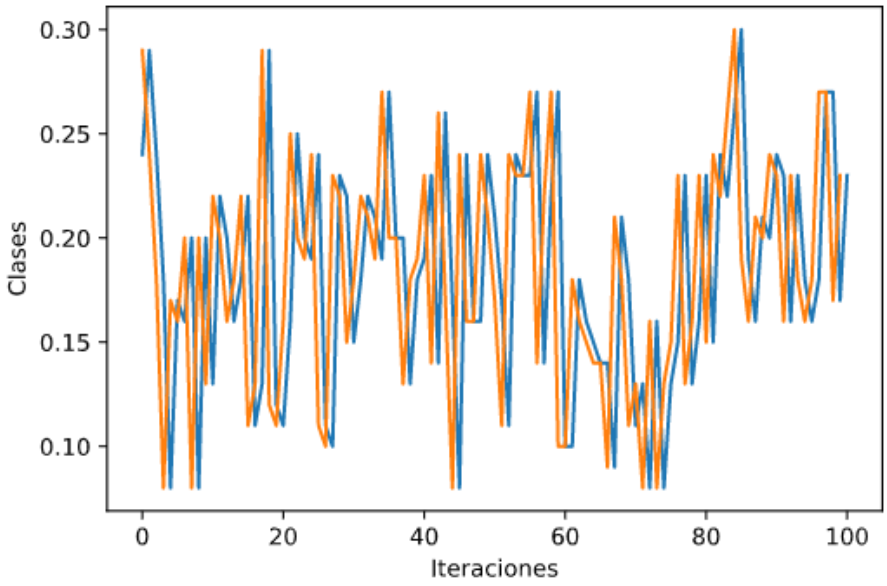

Figura 4.7: Predicción de 100 segmentos (desfasada por un segmento) basada en seleccionar los elementos 1001 al 1005.

Algo interesante a observar, es corroborar el comportamiento del modelo dados los últimos 5 elementos de la base de datos. Lo que ocurre es que el modelo comienza a repetir periódicamente un patrón al infinito. Contrario a esto, al proporcionarle los

5 antepenúltimos elementos, recrea exactamente la misma secuencia de datos en el orden original.

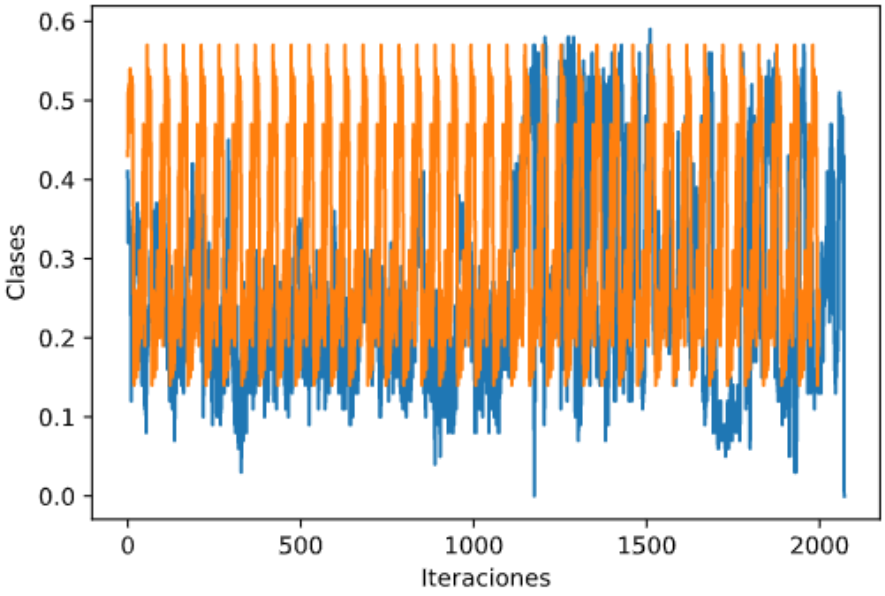

Figura 4.8: Predicción de toda la secuencia original (desfasada por 20 segmentos) basada en seleccionar los últimos cinco elementos de la base de datos.

Arquitectura de la red LSTM utilizada para estos ejemplos:

Arquitectura de la red LSTM

```
1
2  def create_network(network_input, n_unique_features):
3      """ estructura de la red neuronal LSTM """
4      model = Sequential()
5      neurons = 64
6      model.add(LSTM(
7          neurons,
8          input_shape=(network_input.shape[1], network_input.shape[2]),
9          recurrent_dropout=0.3,
10         return_sequences=True))
11     model.add(LSTM(neurons, return_sequences=True, recurrent_dropout
       =0.3))
12     model.add(LSTM(neurons, return_sequences=None))
13     model.add(BatchNorm())
14     model.add(Dropout(0.3))
15     model.add(Dense(neurons))
```

```
16      model.add(Activation('relu'))
17      model.add(BatchNorm())
18      model.add(Dropout(0.3))
19      model.add(Dense(n_unique_features))
20      model.add(Activation('softmax'))
21      model.compile(loss='categorical_crossentropy', optimizer='rmsprop'
        )
22      return model
```

IOFeedback

IOFeedback es una improvisación construida con la forma *ABA* en la cual, empleo, *a grosso modo*, cinco elementos sonoros: feedback, vibraciones con resortes, silencios, paisaje sonoro circundante (perros ladrando) y resonancias acústicas producidas por la tensión y distensión de la membrana de una bocina amplificada con micrófonos de contacto, la cual intervine con algunos tornillos de diferentes tamaños. Además utilicé resortes tensados que producían un efecto tipo *spring reverb*, el cual me permitió aprovechar la estela que dejaba como elemento de contraste. Aquí el enlace para escuchar esta improvisación:

`https://archive.org/details/io-feedback`

Esta improvisación fue segmentada en 751 partes por el algoritmo SBIC, de ellas extraje el descriptor audio aplanamiento espectral, después, los datos entregados por Essentia fueron redondeados a dos dígitos. Este procedimiento, generó un total de 40 elementos diferentes que aparecen en diversas ocasiones a lo largo de la secuencia analizada, la cual fue distribuída en 5 pasos de tiempo.

Pese a que los datos fueron entrenados por esta red un total de 500000 épocas, el modelo no llegó a predecir al cien por ciento la totalidad de la improvisación. Una hipótesis fue que si se entrenaba más veces el modelo, podría llegar a predecir la totalidad de la secuencia de datos originales, ya que la gradiente de descenso solamente había alcanzado a llegar a 0.0115. De acuerdo a los resultados obtenidos con ejemplos como el *Movil II* o *Sink into Return* es posible llegar a la conclusión de que un modelo basado en redes neuronales LSTM, es capaz de hacer predicciones muy acertadas y recrear de manera completamente fiel la improvisación solo cuando la gradiente de

descenso ha alcanzado a llegar a 0.001 o aproximado. Después de seguir entrenando durante tres horas más el modelo, la gradiente de descenso no logró traspasar más abajo de 0.0111. Esto significa que el modelo había llegado al punto de convergencia, donde no puede aprender más de los datos presentados. Derivado de este análisis, es probable que este resultado se deba a la complejidad de las secuencias de IOFeedback y la poca cantidad de datos para hacer una generalización. Después de varias pruebas, revisar errores y modificar la arquitectura de la red, llegué a la conclusión de que el modelo es incapaz de generalizar los datos presentados debido a la poca redundancia en los datos detectados por la red LSTM. A continuación, presento los resultados de la reconstrucción de IOFeedback que realizó la red:

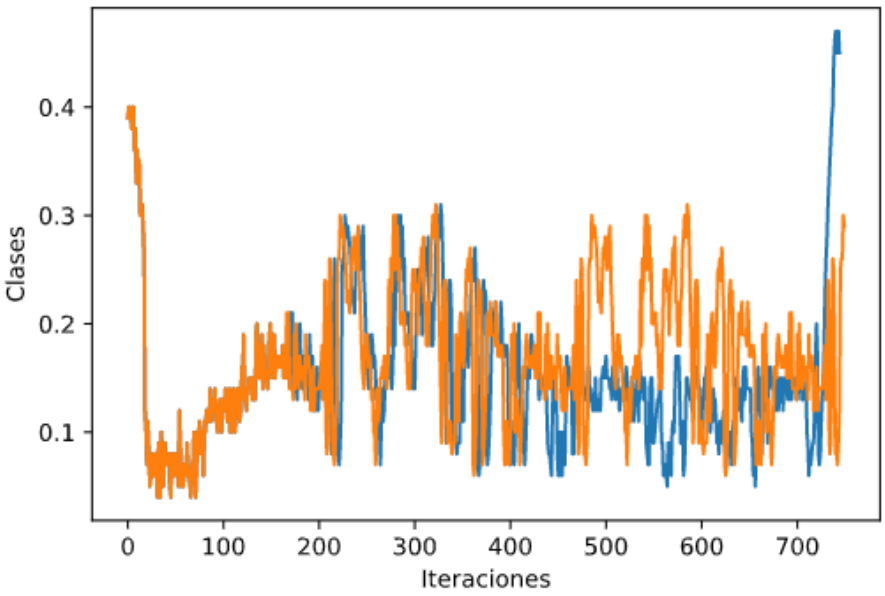

Figura 4.9: En azul 750 segmentos originales de IOFeedback, en naranja la predicción del modelo LSTM.

Como podemos ver en la imagen anterior, el modelo es capaz de reconstruir poco más de la primera mitad de la improvisación pero en el segmento 400 comienza a divergir. Un problema observable en este caso, es que pese a la cantidad de épocas de entrenamiento por las que pasó la red LSTM, llega al punto de convergencia donde no hay un mayor descenso en la gradiente y el modelo se atasca en un solo valor, obstruyendo su posibilidad de aprender más sobre la estructura de datos de entrada e imposibilitando su capacidad para encontrar un punto de mayor optimización.

Es común que en las exploraciones que he llevado a cabo hasta el momento, surja este problema de obstrucción en la gradiente de descenso al entrenar una red LSTM con ejemplos de audio mucho más complejos que los descritos en el apartado anterior. Vayamos a analizar porqué sucede esto y qué opciones, configuraciones y modificaciones sería necesario realizar para poder formalizar en términos numéricos improvisaciones con una menor o igual redundancia en sus datos que IOFeedback.

Uno de estos cambios consistió en aumentar el tamaño de las series que le son entregadas a la red LSTM de 5 a 20 pasos de tiempo. Esto generó un resultado completamente diferente en el cual es posible observar una reconstrucción literal de la secuencia original pese a la poca redundancia.

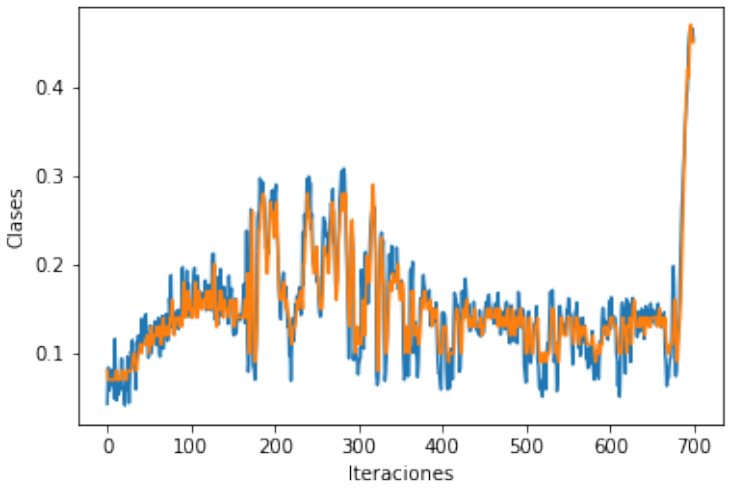

Figura 4.10: En azul 751 segmentos originales de IOFeedback, en naranja la predicción del modelo LSTM al proporcionarle 20 pasos de tiempo.

A este respecto cabe preguntarse, ¿qué papel juega la complejidad de una improvisación versus la cantidad de pasos de tiempo que se le asigna a la red LSTM? Al parecer, la relación cantidad de elementos por pasos de tiempo y variabilidad de las series numéricas son dependientes en sí mismas, vayamos a comprobar que sucede en el ejemplo de *Provenance Unknow 2* de Derek Bailey el cual fue cortado por el algoritmo de segmentación en 739 fragmentos.

Provenance Unknow 2: resultados basados en 5 pasos de tiempo

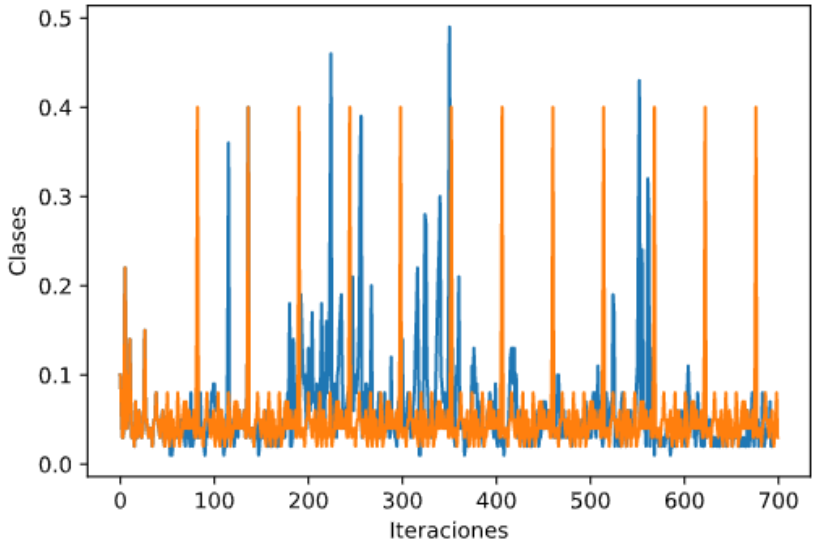

Figura 4.11: Predicción de toda la secuencia original basada en 5 pasos de tiempo, seleccionando los elementos que van del 1 al 5.

Provenance Unknow 2: resultados basados en 20 pasos de tiempo

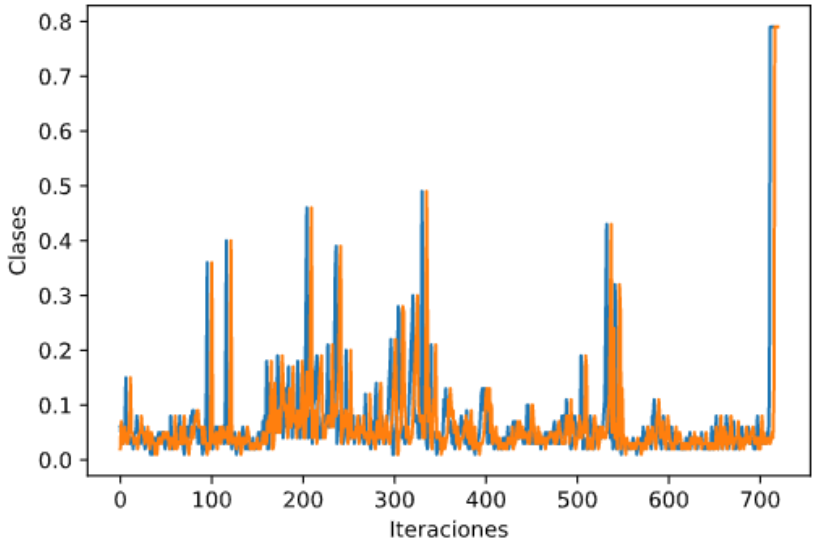

Figura 4.12: Predicción de toda la secuencia original basada en 20 pasos de tiempo, seleccionando los elementos que van del 1 al 20.

Provenance Unknow 2: resultados basados en 100 pasos de tiempo

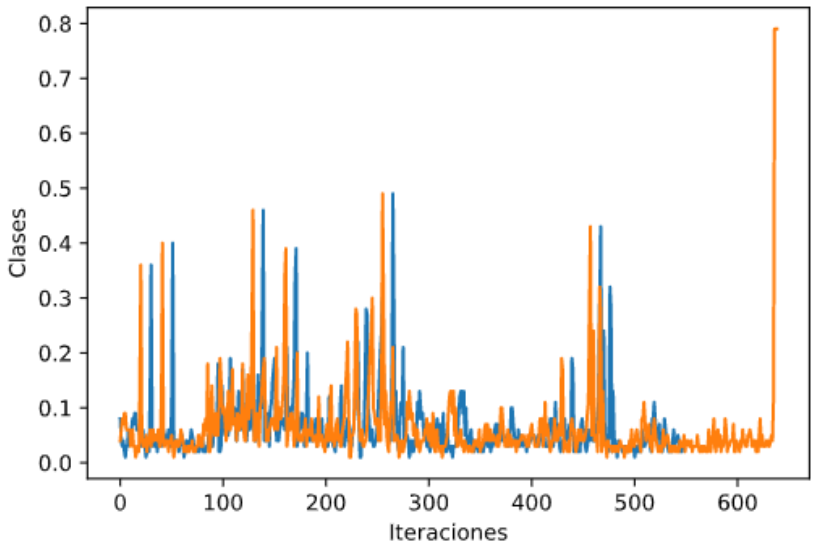

Figura 4.13: Predicción de toda la secuencia original basada en 100 pasos de tiempo, seleccionando los elementos que van del 1 al 100.

Provenance Unknow 2: resultados basados en 300 pasos de tiempo

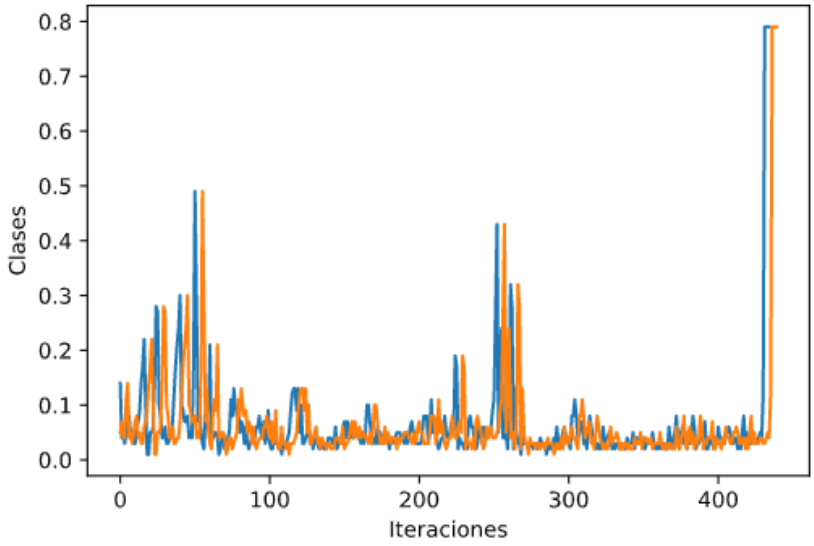

Figura 4.14: Predicción de toda la secuencia original basada en 300 pasos de tiempo, seleccionando los elementos que van del 1 al 300.

Como se observa en las figuras anteriores, un fenómeno similar a IOFeedback ocurre con Provenance Unkown 2, dado que al intentar reconstruir la secuencia basada

en 5 pasos de tiempo el modelo cae en un tipo de bucle que poco se asemeja a la secuencia original. Tomando como punto de partida los 20, 100 y 300 pasos de tiempo el resultado es completamente diferente, pudiendo predecir toda la secuencia original en todos los casos. Lo que concluye de aquí es que la información tiene tanta variabilidad en sus secuencias que le resulta imposible al modelo generalizar sus predicciones usando 5 pasos de tiempo y que necesita secuencias de mayor tamaño o una mayor cantidad de datos para lograr resolver la tarea planteada.

4.3. LSTM secuencia a secuencia

Una vez realizados estos experimentos, el siguiente desarrollo fue dotar a SEALI de herramientas que le permitieran hacer predicciones multivariable al entregarle vectores compuestos de varias características de audio. Antes de encontrar una solución certera a este problema, un largo proceso de investigación heurística y documental fue necesario para determinar cómo entrenar certeramente un modelo LSTM *secuencia a secuencia* (con variables de entrada y predicciones múltiples). Al realizar las primeras pruebas, tuve serios problemas de entrenamiento, y más aún, al intentar replicar la estructura de la improvisación (incluso usando los mismos datos de entrada con los que fue entrenado el modelo). Con todo ello, determiné que era imprescindible integrar otras funciones (capas) que permitieran trabajar y obtener predicciones certeras dado un formato multivariable a la red. Estas capas son, *Repeat Vector* (repetición de vectores) y *Time distributed* (tiempo distribuido). Aplicar éstas capas representó un cambio total respecto a los primeros experimentos realizados al intentar predecir múltiples variables con una red LSTM. Ello permitió integrar a SEALI funciones para predecir estructuras multiparamétricas y descifrar los momentos de una improvisación libre más allá de lo que ocurre al predecir un solo vector como vimos en los apartados anteriores.

Capa distribuida en el tiempo

Si bien algunos autores mencionan que las redes LSTM son muy potentes, para poder explorar ese potencial es necesario determinar en qué momentos se requieren funciones como la repetición de vectores, el aplanamiento de datos, las capas de dis-

tribución temporal sencillas y las capas completamente conectadas, etc. Determinar si es pertinente colocar estas capas antes o después de la red LSTM resulta bastante confuso. Todo ello sin mencionar, que las descripciones de estas funciones y los ejemplos que hay para su implementación son un poco crípticas. Por ejemplo, la capa del método *TimeDistributedDense* es descrita como un contenedor o envoltura *(*wrapper) de la capa inicial que permite aplicar capas subsecuentes a cada segmento temporal de las entradas de la red.[12] Dado este contexto resulta complicado determinar como usar estas capas, cuáles son sus parámetros internos, dónde colocarlas, cuántas utilizar, así como su pertinencia para utilizarlas o no, etc. La confusión en su implementación siguió en aumento cuando intenté aprender a usar estas capas contenedoras especiales a través de los foros de discusión en la red como *StackOverflow*, *Github* o *Keras*.

El tutorial que me ayudó a comprender cómo aplicar y usar estas redes contenedoras lo encontré en el libro *Long Short-Term Memory Networks With Python* del investigador Jason Brownlee (Brownlee, 2017). Desde una perspectiva sencilla, provee ejemplos que pude adaptar a mi problema específico y así acotar las dificultades a las que me enfrenté en la implementación de estas redes.

En el capítulo *El codificador y decodificador LSTM, Predicción de problemas secuencia a secuencia* Brownlee describe que la predicción de secuencia a secuencia o *seq2seq* implica predecir los siguientes valores dada una secuencia de entrada de valores. Este problema también es conocido como problema de predicción de secuencias de tipo *muchos a muchos many to many*. Esto generalmente implica el uso de múltiples pasos de tiempo de entrada (es decir, lo que entiendo que hace la capa del método *TimeDistributedDense*), para poder predecir la siguiente secuencia. De modo que muchas variables con muchos pasos de tiempo pueden ser analizadas y la salida que entrega el modelo sería la siguiente predicción en forma de secuencia multiparamétrica. Es importante recalcar que para un correcto funcionamiento del modelo de predicción utilizando esta capa, el número de pasos de tiempo y dimensiones para entrenar el modelo, tiene que corresponder con los datos que se le entregarán al modelo para generar nuevas predicciones. Por ejemplo, si el modelo fue entrenado con

[12] `https://keras.io/api/layers/recurrent_layers/time_distributed/`

100 pasos de tiempo de 16 dimensiones, estos datos deben coincidir al momento de solicitar predicciones al modelo.

Para conocer a nivel técnico más sobre estos métodos de las redes LSTM, recomiendo arduamente leer con detenimiento el tutorial *How to Use the TimeDistributed Layer in Keras*[13] donde Brownlee explica a detalle el proceso para predecir secuencias temporales *una a una y muchas a muchas*. Además recomiendo leer el capítulo 9 de su libro *Long Short-Term Memory Networks With Python* (Brownlee, 2017) y el artículo *Sequence to Sequence Learning with Neural Networks* sobre predicción de secuencias aplicado a traducciones en distintos idiomas. (Sutskever *et al.*, 2014)[14]

4.3.1. Resultados del análisis multiparamétrico de la red LSTM

A través de esta investigación he comprobado que las redes LSTM son capaces de recrear estructuras numéricas a varios niveles de profundidad y de varias dimensiones, en algunos casos de improvisaciones individuales, logrando un cien por ciento de certeza dada la secuencia original del audio con el que fue entrenado el modelo. Esto es posible una vez que la red LSTM llegó al punto de convergencia con una métrica de pérdida cercana a cero. Para comprobar este experimento, a continuación muestro de la figura 4.15 a la figura 4.20 tres segmentos diferentes de una improvisación libre tocada por el improvisador y violinista Fabian Rangel acompañados de las reconstrucciones producidas por la red LSTM.

Estos resultados podrían parecer muy obvios para un experto en redes neuronales ya que dada la misma secuencia original con la que el modelo fue entrenado, éste debería de regresar los mismos datos de la serie original. Sin embargo, en el apartado *Recomposición automática en tiempo diferido con LSTM* de este mismo capítulo, pudimos observar que en algunos casos fue imposible reconstruir fielmente las secuencias debido a la complejidad de las muestras. Llegar a reconstruir fielmente estos resultados multivariable, no fue una tarea fácil ya que el proceso implicó un profundo trabajo heurístico para determinar qué parámetros de la arquitectura de la red LSTM eran óptimos, y cómo formatear de manera correcta los datos de en-

[13] `https://bit.ly/3FiCHJJ` Fecha de consulta 17 de noviembre 2021
[14] `https://arxiv.org/pdf/1409.3215.pdf` Fecha de consulta 14 de Noviembre 2021.

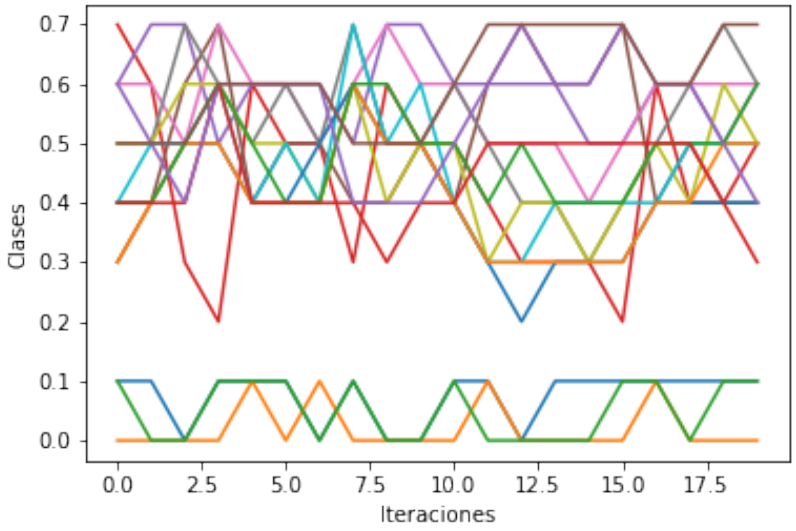

Figura 4.15: Segmentos originales 80 al 100 de la improvisación de Fabian Rangel.

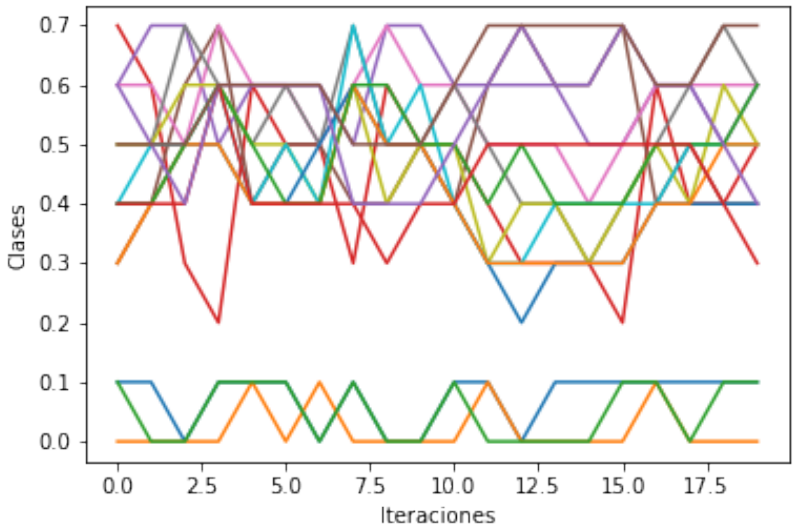

Figura 4.16: Predicción de la secuencia original basada en 20 pasos de tiempo, seleccionando los segmentos que van del 60 al 80. La predicción esperada va del segmento 80 al 100

trenamiento para la red. Uno de ellos consistió, como mencioné en el apartado de *Recomposición automática en tiempo diferido con LSTM*, en reducir la complejidad de la base de datos al redondear los valores de los datos a 2 y 3 dígitos. De manera

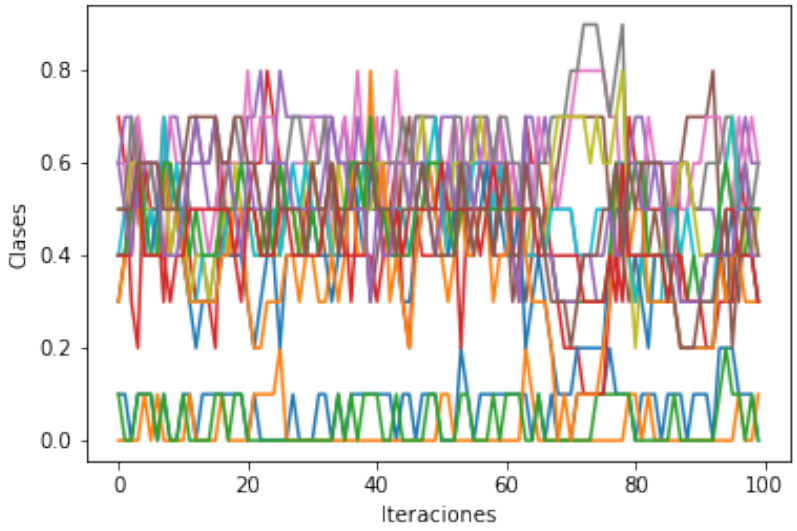

Figura 4.17: Segmentos originales 20 al 120 de la improvisación de Fabian Rangel.

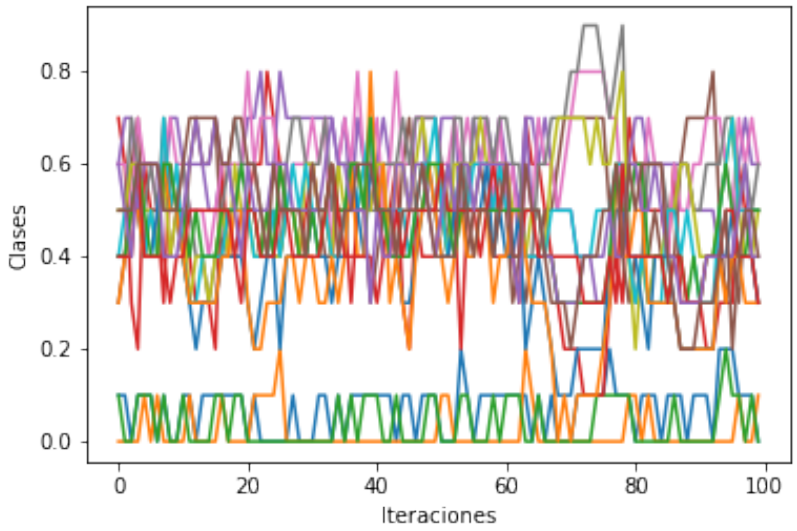

Figura 4.18: Predicción de la secuencia original basada en 20 pasos de tiempo. La predicción esperada va del segmento 20 al 120.

tal que la red LSTM tuviera 11 clases (del 0 al 10) al usar 2 dígitos y 101 clases (del 0 al 101) al reducir a 3 dígitos.

Cabe resaltar que utilizar un método de validación propio del aprendizaje de máquina como *precisión, gradiente de descenso, validación cruzada, etc.* para calcu-

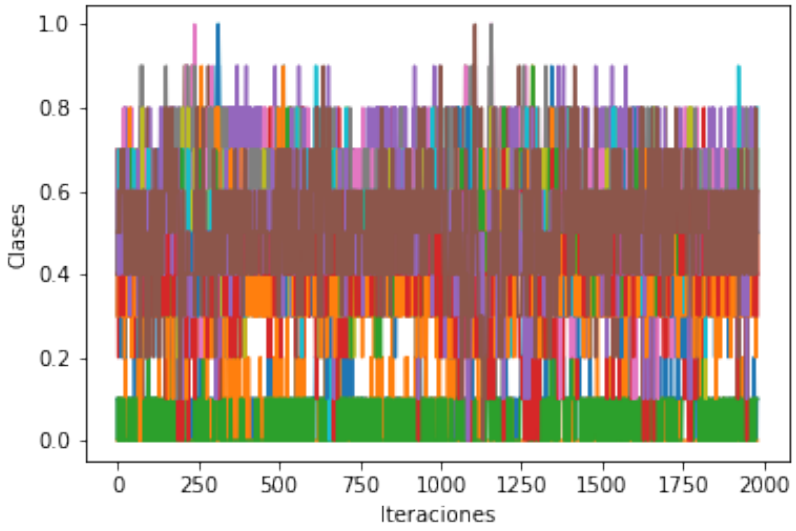

Figura 4.19: Estructura compuesta de 2000 segmentos de la improvisación de Fabian Rangel.

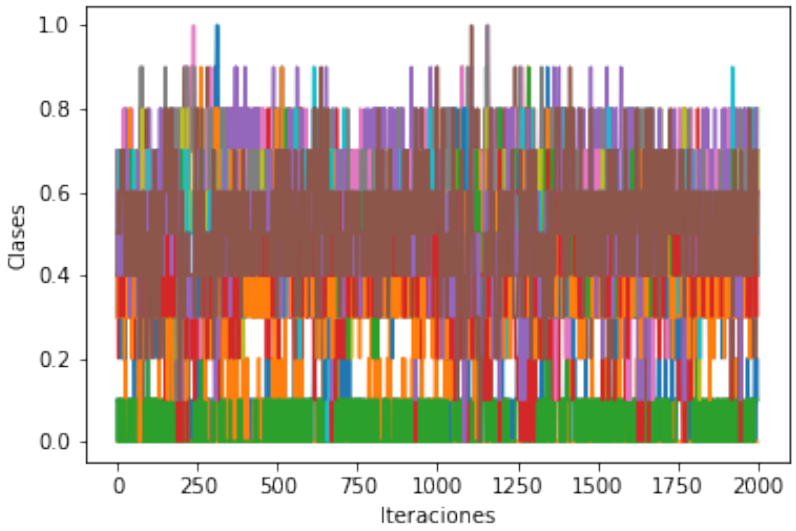

Figura 4.20: Predicción de 2000 segmentos dada la secuencia original basada en 20 pasos de tiempo, tomando los segmentos que van del 0 al 20

lar la certeza del modelo, no necesariamente apunta hacia la validación de un índice de certeza en términos musicales. Determinar qué tan bien o qué tan mal el modelo predice la estructura de una improvisación libre, puede incluso implicar apreciaciones subjetivas tan diversas como personas haya. Validar con indicadores del aprendizaje

de máquinas una manifestación artística sería como caricaturizar la propia realidad. Por una parte, determinar cuál es el rendimiento del modelo para describir un fenómeno tan complejo como la predicción de estructuras en la improvisación libre, implica partir de una mirada reduccionista que limita considerablemente el amplio espectro de posibilidades que no parten de un paradigma estadístico para validar si el sistema de clasificación o predicción está teniendo un buen desempeño. Además, dado que en esta aproximación se dejaron fuera muchas capas de información presentes en la música en general, y en la improvisación libre en particular, depender de este indicador no sería suficiente para validar el rendimiento del sistema. Por ello, para validar la certeza del modelo decidí escuchar las reconstrucciones sonoras que SEALI propone al recibir información que fue utilizada en el proceso de entrenamiento del modelo computacional.

4.3.2. Análisis multivariable y reconstrucción de String Theory

Un siguiente experimento que realicé consistió en dividir en dos partes el disco *String Theory* del improvisador Derek Bailey[15] donde una de las improvisaciones fue reservada para las pruebas y las demás para entrenar al modelo. Después de extraer las características de las improvisaciones de este disco, el conjunto de prueba fue utilizado para comprobar el rendimiento del modelo al intentar recrear la estructura original de la improvisación faltante. En las figuras 4.21 y 4.22 muestro gráficamente los resultados obtenidos al intentar hacer las predicciones sobre datos nunca antes vistos:

Pese a que el modelo convergió y es capaz de recrear secuencias muy similares a las originales dada una secuencia original inicial como se muestra en las figuras 4.23 y 4.24, evidentemente iba resultar extremadamente difícil que el modelo recreara de manera fiel una secuencia nunca antes vista debido a la amplia cantidad de parámetros y diferencias de una improvisación a otra. En la figura 4.23 vemos cien segmentos originales y en la figura 4.24, el intento por reconstruir con base en predicciones la misma serie. El modelo debería recrear fielmente estos datos, sin embargo, pese a que los datos están en su base de datos, no logró replicar la secuencia original de forma

[15] `https://www.discogs.com/release/1039258-Derek-Bailey-String-Theory`

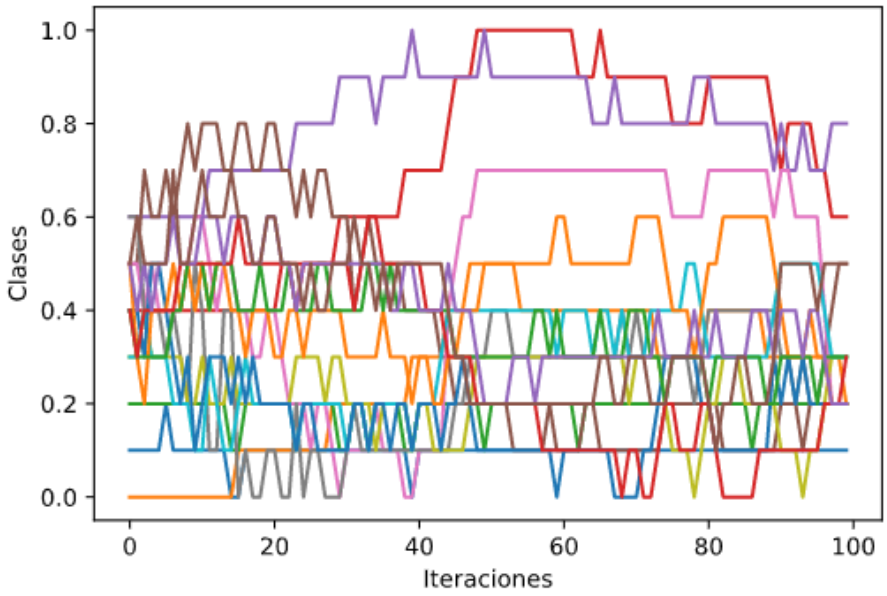

Figura 4.21: 100 segmentos originales (200 al 300) de la improvisación no contemplada en el entrenamiento del modelo.

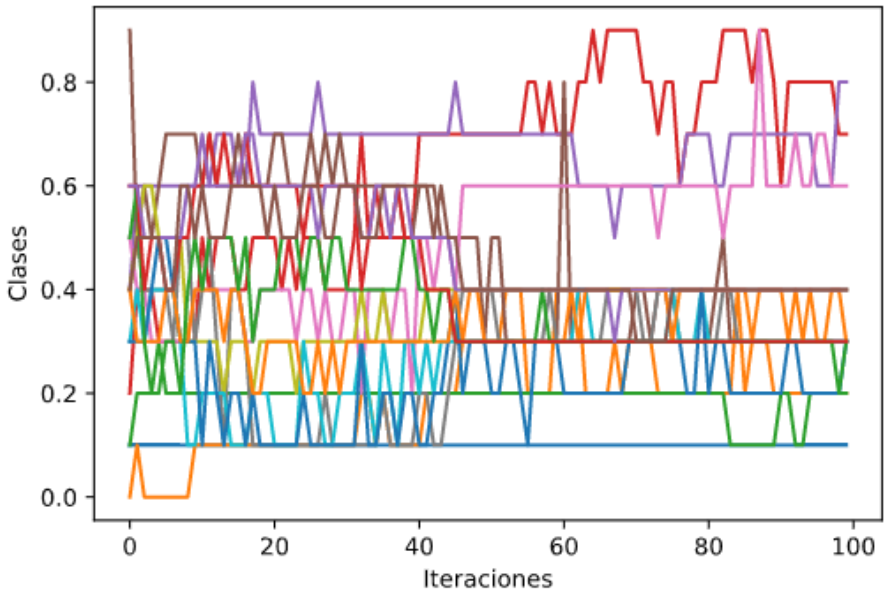

Figura 4.22: 100 predicciones multivariable de datos dada la primera secuencia de 100 valores nunca antes vista por el modelo. Se esperaría obtener la secuencia de la figura 4.21.

cien por ciento certera. Para que el modelo pudiera replicar sus propias secuencias de entrenamiento, los datos necesitaban mayor redundancia, así opté por cuadruplicar la base de datos que en un inicio era de 4248 muestras y terminé con un total de

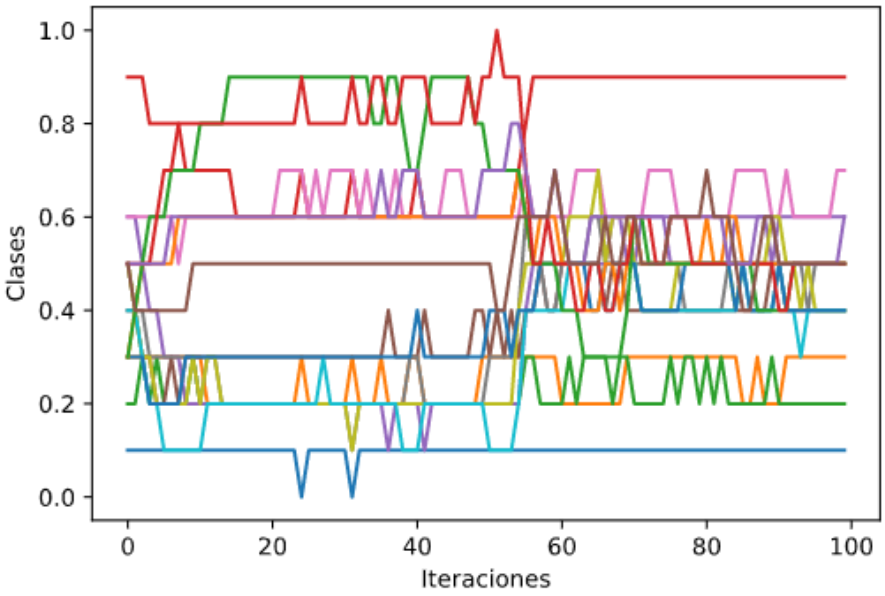

Figura 4.23: Serie de 100 segmentos originales multivariable.

16992 muestras. Esto significó un cambio sustancial como puede ser apreciado en las figuras 4.25 y 4.26.

Pese a la similaridad que hay entre la improvisación que no fue ocupada para entrenar al modelo con la similaridad estética de las otras improvisaciones contempladas en el proceso de entrenamiento resulta sumamente difícil poder llegar a una predicción cien por ciento fiel. Esto no significa un problema, si bien la predicción no es certera, partiendo de las figuras 4.21 y 4.22, es posible observar cierta similaridad, en cuanto a actividad y dirección, en las secciones que componen estos cien segmentos. Por otro lado, terminamos con una especie de "huella digital" de la esencia aproximativa de Bailey que le permite al modelo generar variaciones en el estilo del improvisador pero pudiendo generar sus propias derivas producto de la propia algoritmicidad.

En las figuras 4.27 4.28 muestro los resultados de predicción del modelo al ser entrenado con los datos cuadruplicados e intentar reconstruir una secuencia nunca antes vista:

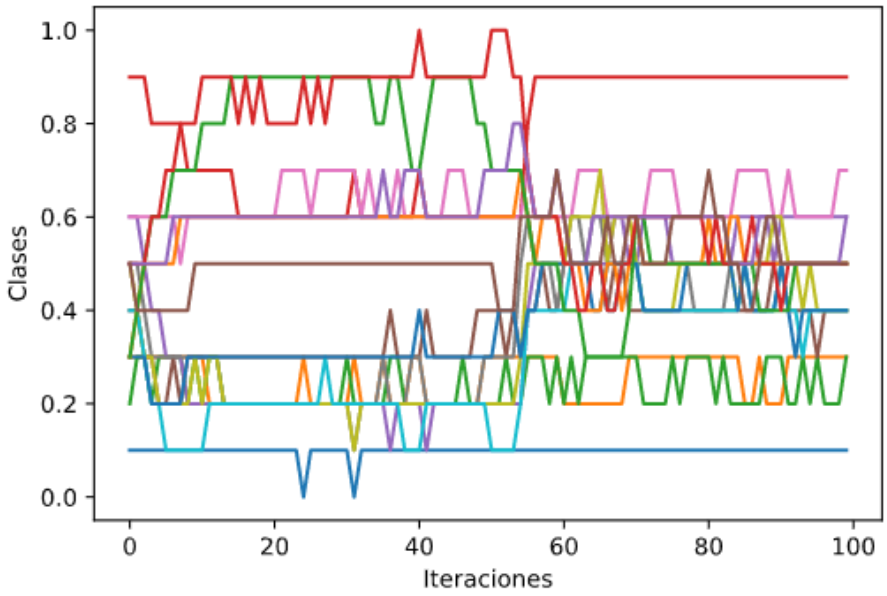

Figura 4.24: Predicción multivariable de segmentos originales tomando 20 pasos de tiempo.

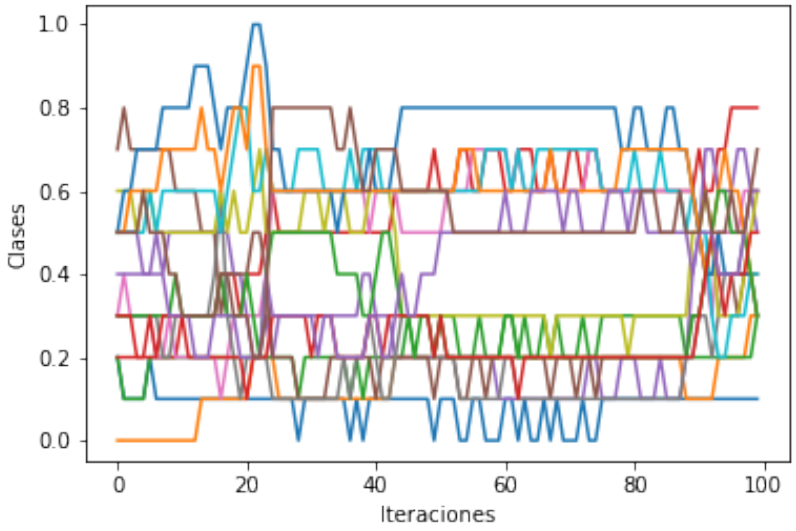

Figura 4.25: Serie de datos originales multivariable al cuadriplicar la base de datos.

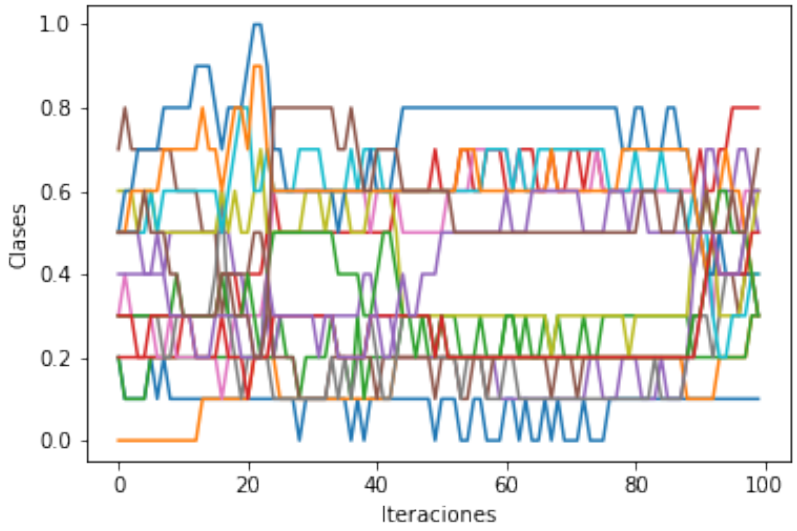

Figura 4.26: 100 Predicciones multivariable con LSTM tomando 20 pasos de tiempo, usando el modelo que aprendió con la base de datos cuadruplicada.

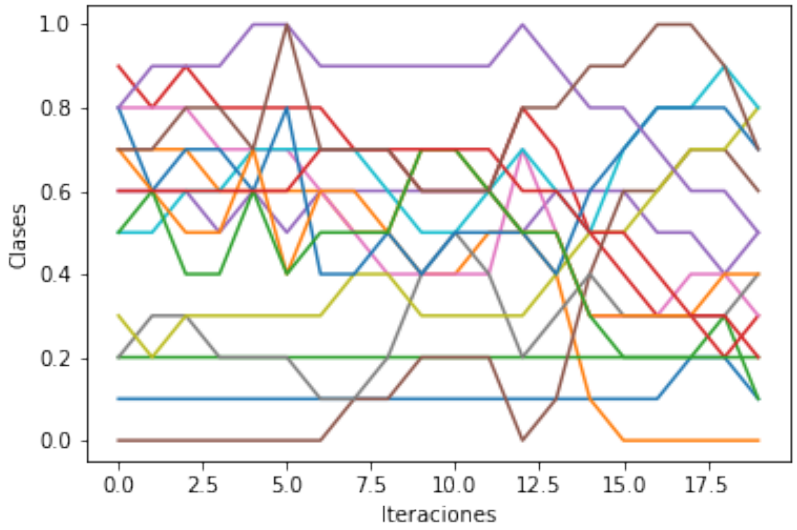

Figura 4.27: Serie de datos originales multivariable al cuadriplicar la base de datos

Reconstrucción de secuencias Derek Bailey

A través del siguiente código realicé las pruebas anteriores:

```
file_in = "datasets/derek_bailey_electric-FCLMFCCs.csv"
load_model = "Models/derek_bailey_electric_FLCMFCC_100TS-256N_242_0
    .0006.hdf5"
```

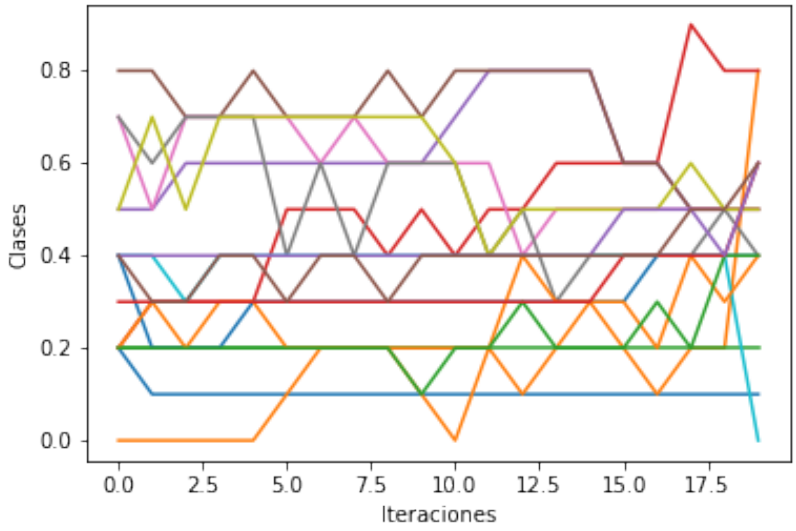

Figura 4.28: 20 Predicciones multivariable con LSTM tomando 20 pasos de tiempo, usando el modelo que aprendió con la base de datos cuadruplicada.

```
3
4
5  def parse_database():
6      trainset = []
7      with open(file_in) as archivo:
8          lineas = archivo.read().splitlines()
9      for l in lineas:
10       linea = l.split(' ')
11       floats = [float(i) for i in linea]
12       trainset.append(floats)
13     return trainset
14
15
16 data = parse_database()
17 min_max_scaler = MinMaxScaler()
18 data_scaled = min_max_scaler.fit_transform(data)
19 data = np.round(data_scaled, 1)
20
21
22 n_steps = 100
23
24
```

```
25 def split_sequences(sequences, n_steps):
26     X, y = list(), list()
27     for i in range(len(sequences)):
28         end_ix = i + n_steps
29         if end_ix > len(sequences)-1:
30             break
31         seq_x, seq_y = sequences[i:end_ix, :], sequences[end_ix, :]
32         X.append(seq_x)
33         y.append(seq_y)
34     return array(X), array(y)
35
36
37 X, y = split_sequences(data, n_steps=n_steps)
38
39
40 y = (y*10)
41 y = y.astype(int)
42 y_categorical = np_utils.to_categorical(y)
43
44
45 def create_network(network_input, neurons):
46     model = Sequential()
47     neurons = neurons
48     model.add(LSTM(
49         neurons,
50         input_shape=(network_input.shape[1], network_input.shape[2]),
51         return_sequences=False
52     ))
53     model.add(RepeatVector(network_input.shape[2]))
54     model.add(LSTM(neurons, return_sequences=True))
55     model.add(TimeDistributed(Dense(y_categorical.shape[2], activation
    ='softmax')))
56     model.compile(loss='categorical_crossentropy' , optimizer='adam',
    metrics=[ 'accuracy' ])
57     print("Debuging", network_input.shape)
58     print("Debuging", y_categorical.shape)
59     model.load_weights(load_model)
60     return model
61
```

```
62
63 inputs = X[50:150]
64 print(inputs)
65 model = create_network(X, neurons=256)
66 prediction = model.predict(inputs, verbose=0)
67
68
69 def set_zero(sample, d, val):
70     argmax_idxs = sample.argmax(d)
71     return argmax_idxs*0.1
72
73
74 prediction = set_zero(prediction, d=2, val=0)
75
76
77 plt.plot(data[150:250])
78 plt.plot(prediction)
79 plt.ylabel('Clases')
80 plt.xlabel('Iteraciones')
81 plt.savefig('TDprediccion.png')
82 plt.show()
```

4.3.3. Aplicaciones de SEALI en tiempo–real

Algunas de las técnicas que he abordado hasta el momento son preliminares, han sido desarrolladas con el mero propósito de estetizar los resultados de las predicciones que hace el modelo en tiempo–real. Una de las propuestas exploradas es que al momento de recibir una secuencia de audio digital de n pasos de tiempo, el modelo regrese una predicción basada en los segmentos de audio de la base de datos con que fue entrenado. Esta predicción es comparada con toda la base de datos de entrenamiento para obtener el caso con menor diferencia entre la predicción generada y los casos analizados en la base de datos original. De manera tal que el sistema genera una especie de respuesta interactiva basada en la forma más cercana en la que, de acuerdo con el contexto aprendido por el modelo, podría desarrollarse la improvisación de forma subsecuente. De este modo, se podría explorar una perspectiva contrastante tomando

el audio más distinto, calculando la mayor diferencia en relación a la predicción que el modelo ha realizado en un momento determinado.

Interacción en tiempo–real mediante la predicción de secuencias de 20 pasos de tiempo en *String Theory* de Derek Bailey

A continuación presento 2 grabaciones que dan cuenta de la interacción de SEALI al utilizar dos modelos simultáneos entrenados para predecir los siguientes 100 y 500 pasos de tiempo de la estructura de la improvisación del álbum *String Theory* de Derek Bailey. SEALI interactúa al lanzar los audios más similares, dentro de su base de datos, a través de las predicciones que hace el modelo, todo ello con base en el desenvolvimiento estructural analizado por los descriptores del estilo de Derek Bailey. En términos técnicos, Python envía mediante un mensaje OSC el audio seleccionado por el modelo, Supercollider lo recibe y se encarga de reproducir el audio aplicando filtros, envolventes y transformaciones temporales.

En las grabaciones, SEALI interactúa al escuchar una improvisación conocida por el modelo (que va del minuto 0:00 al 04:48) y después mediante la improvisación nunca antes vista por el modelo (del minuto 4:48 a 10:16). En la primera grabación, muestro una mezcla de la improvisación de Bailey acompañado por SEALI. En la segunda, presento la interacción de SEALI abstraída del contexto sonoro que la estimula. Es importante resaltar que no hay interacción con el espacio acústico, todo sucede dentro de las conexiones internas de la señal de audio digital. Si bien, SEALI podría escuchar al entorno acústico mediante un micrófono y a sí misma mediante conexiones internas de audio, para estos resultados, SEALI no escucha lo ella misma produce, es "agnóstica" a su propuesta sonora. Esto lo propuse con el fin de tener menos elementos interactuando entre sí y poder escuchar la propuesta que SEALI genera desde un análisis acotado. De esta manera solo atiende al estímulo de la improvisación de Bailey. Finalmente, cabe señalar que los elementos sonoros de la improvisación nunca antes vista no están presentes en la base de datos de audio que SEALI toma para la interacción sonora.

SEALI reaccionando a la improvisación nunca antes vista de Derek Bailey + más la respectiva improvisación

https://archive.org/details/seali-bailey-20-ts

SEALI reaccionando a la improvisación nunca antes vista de Derek Bailey + aislando el estímulo sonoro de la mezcla

https://archive.org/details/seali-bailey-20-ts/SEALI--Bailey_20_TS.wav

Video de la improvisación

https://youtu.be/fEmSINVHY5k

A continuación presento los códigos de Python y Supercollier que me permitieron generar las grabaciones:

Extracción de características en tiempo–real, predicción y envío de datos vía OSC

```
min_max_scaler = MinMaxScaler()
sampleRate = 44100
frameSize = 2048
hopSize = 2048
numberBands = 13
onsets = 1
loudness_bands = 1
patchSize = 1
bufferSize = patchSize * hopSize
buffer = np.zeros(bufferSize, dtype='float32')
vectorInput = VectorInput(buffer)
frameCutter = FrameCutter(frameSize=frameSize, hopSize=hopSize)
w = Windowing(type = 'hann')
spec = Spectrum()
mfcc = MFCC(numberCoefficients=numberBands)
fft = FFT()
c2p = CartesianToPolar()
onset = OnsetDetection()
eqloud = EqualLoudness()
```

```
flatness = Flatness()
envelope = Envelope()
accu = RealAccumulator()
loudness = Loudness()
complexity = SpectralComplexity()
centroid = Centroid()
square = UnaryOperator(type='square')

client = udp_client.SimpleUDPClient('127.0.0.1', 5006)
load_model = "Models/derek_bailey_electric_FLCMFCC_100TS-64N_242_0
    .0006.hdf5"
json_in = "datasets/derek_bailey_electric-FCLMFCCs.json"

def multifeaturesExtractor():
    pool = Pool()
    vectorInput.data  >> frameCutter.signal
    frameCutter.frame >> w.frame >> spec.frame
    spec.spectrum     >> flatness.array
    frameCutter.frame >> loudness.signal
    spec.spectrum     >> centroid.array
    spec.spectrum     >> mfcc.spectrum
    flatness.flatness >> (pool, 'flatness')
    loudness.loudness >> (pool, 'loudness')
    centroid.centroid >> (pool, 'centroid')
    mfcc.mfcc         >> (pool, 'mfcc')
    mfcc.bands        >> None
    return pool

pool = multifeaturesExtractor()

def callback(data):
    buffer[:] = array(unpack('f' * bufferSize, data))
    flatnessBuffer = np.zeros([1])
    loudnessBuffer = np.zeros([1])
    centroidBuffer = np.zeros([1])
    mfccBuffer = np.zeros([numberBands])
    reset(vectorInput)
    run(vectorInput)
    flatnessBuffer = np.roll(flatnessBuffer, -patchSize)
```

```
58      loudnessBuffer = np.roll(loudnessBuffer, -patchSize)
59      centroidBuffer = np.roll(centroidBuffer, -patchSize)
60      mfccBuffer = np.roll(mfccBuffer, -patchSize)
61      flatnessBuffer = pool['flatness'][-patchSize]
62      loudnessBuffer = pool['loudness'][-patchSize]
63      centroidBuffer = pool['centroid'][-patchSize]
64      mfccBuffer = pool['mfcc'][-patchSize]
65      features = np.concatenate((flatnessBuffer, loudnessBuffer,
        centroidBuffer, mfccBuffer), axis=None)
66      features = features.tolist()
67      return features
68
69  def create_network():
70      model = Sequential()
71      neurons = 256
72      model.add(LSTM(
73          neurons,
74          input_shape=(100, 16),
75          return_sequences=False
76      ))
77      model.add(RepeatVector(16))
78      model.add(LSTM(neurons, return_sequences=True))
79      model.add(TimeDistributed(Dense(11, activation='softmax')))
80      model.compile(loss='categorical_crossentropy' , optimizer='adam',
        metrics=[ 'accuracy' ])
81      model.load_weights(load_model)
82      return model
83
84  min_max_scaler = MinMaxScaler()
85  all_features = []
86  model = create_network()
87
88  def reducer( features, acc, fileData):
89    fileData_ =  np.array(fileData['allFeatures'])
90    features_ = np.array(features)
91    diff = np.linalg.norm(fileData_ - features_)
92    if acc==None:
93      return tz.assoc(fileData, 'diff', diff)
94    else:
```

```
95      if acc['diff'] <= diff:
96          return acc
97      else:
98          return tz.assoc(fileData, 'diff', diff)
99
100 def getClosestCandidate(features):
101       with open(json_in) as f:
102       jsonData = json.load(f)
103       return reduce(lambda acc, fileData: reducer(features, acc,
      fileData), jsonData, None)
104
105 def set_zero(sample, d):
106     argmax_idxs = sample.argmax(d)
107     return argmax_idxs*0.1
108
109 file_selected = '000000'
110
111 def accFeatures(data):
112     global model
113     global all_features
114     global file_selected
115     i = len(all_features)
116     if i == 20:
117         all_features = np.array(all_features)
118         features_scaled = min_max_scaler.fit_transform(all_features)
119         features_scaled = np.round(features_scaled, 1)
120         features_reshaped = np.reshape(features_scaled, (1,
      features_scaled.shape[0], features_scaled.shape[1]))
121         prediction = model.predict(features_reshaped, verbose=0)
122         prediction = set_zero(prediction, d=2)
123         print("Current Model Prediction: \n", prediction)
124         result = list(map(getClosestCandidate, prediction))
125         print("res:",result)
126         print ("File Selected from Dataset:", list(map(lambda x: x['
      file'], result)))
127         all_features = []
128         file_selected = list(map(lambda x: x['file'], result))
129         file_selected = str(file_selected)[1:-1]
130         file_selected = file_selected.replace("'", "")
```

```
131         return client.send_message("/file_selected", file_selected)
132     else:
133         all_features.append(data)
134 
135 with sc.all_microphones(include_loopback=True)[-1].recorder(samplerate
    =sampleRate) as mic:
136   while True:
137       accFeatures(callback(mic.record(numframes=bufferSize).mean(axis
    =1)))
```

Código de Supercollider que recibe los mensajes OSC de Python y reproduce los audios de las predicciones del modelo

```
1 ~bailey = Array.new;
2 ~folder = PathName.new("/media/atsintli/Archive/SEALI/audio
    /segments_bailey_electric");
3 ~folder.entries.do({
4     arg path;
5     ~bailey = ~bailey.add(Buffer.read(s, path.fullPath));
6 });
7 
8 ~seali = NetAddr("127.0.0.1"); // loopback ----
9 OSCdef (\osc_guitar, {|msg, time, addr, recvPort|
10     var file_selected, file;
11     # file_selected, file = msg;
12     file = file.asInteger.postln;
13 
14     if(file > 0){
15         Ndef(\buf,
16             {
17                 var sig, env, rev;
18                 sig = PlayBuf.ar(1, ~bailey[file],
    [-24,-12,0,12,24].choose.midiratio, 1, doneAction: 0);
```

```
19                  env = EnvGen.kr(Env.new([0, 0.8, 0.8, 0],
    [0.1, [0.1,0.2,0.3,0.5,0.7,1.3,2.1,3,4,5].choose, 0.5]))
    ;
20                  sig = RHPF.ar(sig * env, LFNoise1.kr(0.2).
    range(100,600), LFNoise1.kr(LFNoise1.kr(1).range(0.5,5))
    .range(0.05,0.99), 0.70);
21                  rev = GVerb.ar(sig, 90, 55, 0.4, 0.4, 40,
    0.7, 0.99, 0.99);
22                  sig = PanB.ar(rev, LFNoise1.kr(2).range
    (-0.5,0.5), LFNoise1.kr(2).range(-0.5,0.5), 0.2);
23                  GVerb.ar(sig, 30, 30, 0.3, 0.3, 40, 0.5,
    0.7, 0.7, 300, 0.1);
24              });
25          };
26 },'/file_selected',recvPort: 5006);
27
28 Ndef(\buf).play
29 Ndef(\buf).fadeTime=3
```

Análisis de la improvisación de Fabian Rangel

En ese ejemplo empleé el modelo previamente entrenado con la improvisación de Fabian Rangel, y desarrollé un código en Python el cual activa la entrada de la señal de audio, analiza el audio de entrada de la tarjeta de audio (este también puede ser el audio interno del sistema mediante el *patchbay* de *jack*), después el algoritmo espera a que el número de pasos de tiempo que requiere el modelo para hacer una predicción sea llenado. Si el modelo requiere 100 pasos de tiempo para poder hacer una predicción certera, habría que esperar este tiempo para enviar el conjunto de segmentos de audio analizados al modelo predictivo. Una vez que estos datos están completos se normalizan y se disponen en el formato de datos que espera el modelo, en este caso se espera una disposición de los datos en 16 dimensiones. Los datos son analizados por el modelo y entrega una nueva predicción la cual será comparada con la base de datos original. De esta comparativa el algoritmo toma el elemento de su

base de datos más similar a su predicción, y lo reproduce en forma de audio mediante el módulo *pydub* de Python.

En el siguiente enlace es posible escuchar el resultado de esta improvisación:

`https://archive.org/details/fabian-seali`

En el siguiente enlace presento el video de una improvisación de SEALI usando y cuatro modelos LSTM entrenados con una base de datos de gestos de Derek Bailey y otra de Fabian Rangel:

`https://youtu.be/EtqiLpLz3gU`

A continuación presento los dos códigos empleados para la resolución de esta tarea, en la cual, primero hay que entrenar y salvar el modelo y después llamarlo y hacer las predicciones:

Entrenamiento del modelo LSTM muchos a muchos

```
file_in = "datasets/RANGEL_flatness_loudness_centroid_mfcc.csv"

def parse_database():
    trainset = []
    with open(file_in) as archivo:
        lineas = archivo.read().splitlines()
    for l in lineas:
      linea = l.split(' ')
      floats = [float(i) for i in linea]
      trainset.append(floats)
    return trainset

data = parse_database()
min_max_scaler = MinMaxScaler()
data_scaled = min_max_scaler.fit_transform(data)
data = np.round(data_scaled, 1)

n_steps = 20

def split_sequences(sequences, n_steps):
    X, y = list(), list()
```

```
22      for i in range(len(sequences)):
23          end_ix = i + n_steps
24          if end_ix > len(sequences)-1:
25              break
26          seq_x, seq_y = sequences[i:end_ix, :], sequences[end_ix, :]
27          X.append(seq_x)
28          y.append(seq_y)
29      return array(X), array(y)
30
31  X, y = split_sequences(data, n_steps=n_steps)
32
33  #convert to one hot encode
34  y = (y*10) #move decimal for categorical transform
35  y_categorical = np_utils.to_categorical(y)
36
37
38  def create_network(network_input, neurons):
39      model = Sequential()
40      neurons = neurons
41      model.add(LSTM(
42          neurons,
43          input_shape=(network_input.shape[1], network_input.shape[2]),
44          return_sequences=False
45      ))
46      model.add(RepeatVector(network_input.shape[2]))
47      model.add(LSTM(neurons, return_sequences=True))
48      model.add(TimeDistributed(Dense(y_categorical.shape[2], activation
        ='softmax')))
49      model.compile(loss='categorical_crossentropy' , optimizer='adam',
        metrics=[ 'accuracy' ])
50
51  def train(model, network_input, network_output, epochs, batch_size):
52      model.summary()
53      history = model.fit(network_input, network_output, epochs=epochs,
        batch_size=batch_size)
54
55  def train_and_save(model, network_input, network_output, epochs,
        batch_size):
```

```
    filepath = "saved_models/weights/movil_FLC_100TS-64N_{epoch:02d}_{loss:.4f}.hdf5"
    checkpoint = ModelCheckpoint(
        filepath,
        monitor='loss',
        verbose=0,
        save_best_only=True,
        mode='min'
    )
    callbacks_list = [checkpoint]
    model.summary()
    history = model.fit(network_input, network_output, epochs=epochs, batch_size=batch_size, callbacks=callbacks_list)

model = create_network(X, neurons=128)
train_and_save(model, X, y_categorical, epochs=100,  batch_size=32)

inputs = X[0:1]
prediction = model.predict(inputs, verbose=0)

def set_zero(sample, d, val):
    argmax_idxs = sample.argmax(d)
    return argmax_idxs*0.1

prediction = set_zero(prediction, d=2, val=0)
```

Predicción y análisis de la estructura en tiempo–real

```
min_max_scaler = MinMaxScaler()
sampleRate = 44100
frameSize = 2048
hopSize = 2048
numberBands = 13
onsets = 1
loudness_bands = 1
patchSize = 1
bufferSize = patchSize * hopSize
buffer = np.zeros(bufferSize, dtype='float32')
vectorInput = VectorInput(buffer)
frameCutter = FrameCutter(frameSize=frameSize, hopSize=hopSize)
```

```
w = Windowing(type = 'hann')
spec = Spectrum()
mfcc = MFCC(numberCoefficients=numberBands)
fft = FFT()
c2p = CartesianToPolar()
onset = OnsetDetection()
eqloud = EqualLoudness()
flatness = Flatness()
envelope = Envelope()
accu = RealAccumulator()
loudness = Loudness()
complexity = SpectralComplexity()
centroid = Centroid()
square = UnaryOperator(type='square')

load_model = "Models/RANGEL_FLCMFCC_20TS-64N_242_0.0006.hdf5"
json_in = "datasets/RANGEL-FCLMFCCs.json"

def multifeaturesExtractor():
    pool = Pool()
    vectorInput.data  >> frameCutter.signal
    frameCutter.frame >> w.frame >> spec.frame
    spec.spectrum     >> flatness.array
    frameCutter.frame >> loudness.signal
    spec.spectrum     >> centroid.array
    spec.spectrum     >> mfcc.spectrum
    flatness.flatness >> (pool, 'flatness')
    loudness.loudness >> (pool, 'loudness')
    centroid.centroid >> (pool, 'centroid')
    mfcc.mfcc         >> (pool, 'mfcc')
    mfcc.bands        >> None
    return pool

pool = multifeaturesExtractor()

def callback(data):
    buffer[:] = array(unpack('f' * bufferSize, data))
    flatnessBuffer = np.zeros([1])
    loudnessBuffer = np.zeros([1])
```

```
52      centroidBuffer = np.zeros([1])
53      mfccBuffer = np.zeros([numberBands])
54      reset(vectorInput)
55      run(vectorInput)
56      flatnessBuffer = np.roll(flatnessBuffer, -patchSize)
57      loudnessBuffer = np.roll(loudnessBuffer, -patchSize)
58      centroidBuffer = np.roll(centroidBuffer, -patchSize)
59      mfccBuffer = np.roll(mfccBuffer, -patchSize)
60      flatnessBuffer = pool['flatness'][-patchSize]
61      loudnessBuffer = pool['loudness'][-patchSize]
62      centroidBuffer = pool['centroid'][-patchSize]
63      mfccBuffer = pool['mfcc'][-patchSize]
64      features = np.concatenate((flatnessBuffer, loudnessBuffer,
        centroidBuffer, mfccBuffer), axis=None)
65      features = features.tolist()
66      return features
67
68  def create_network():
69      model = Sequential()
70      neurons = 256
71      model.add(LSTM(
72          neurons,
73          input_shape=(20, 16),
74          return_sequences=False
75      ))
76      model.add(RepeatVector(16))
77      model.add(LSTM(neurons, return_sequences=True))
78      model.add(TimeDistributed(Dense(11, activation='softmax')))
79      model.compile(loss='categorical_crossentropy' , optimizer='adam',
        metrics=[ 'accuracy' ])
80      model.load_weights(load_model)
81      return model
82
83  min_max_scaler = MinMaxScaler()
84  all_features = []
85  model = create_network()
86
87  def reducer( features, acc, fileData):
88    fileData_ =  np.array(fileData['allFeatures'])
```

```
89    features_ = np.array(features)
90    diff = np.linalg.norm(fileData_ - features_)
91    if acc==None:
92      return tz.assoc(fileData, 'diff', diff)
93    else:
94      if acc['diff'] <= diff:
95          return acc
96      else:
97          return tz.assoc(fileData, 'diff', diff)
98
99  def getClosestCandidate(features):
100       with open(json_in) as f:
101       jsonData = json.load(f)
102       return reduce(lambda acc, fileData: reducer(features, acc,
      fileData), jsonData, None)
103
104 def playAudio(file, time):
105     loop = AudioSegment.from_wav(file + ".wav")
106     loop2 = loop * 1
107     length = len(loop2)
108     print(length)
109     fade_time = int(length * 0.3)
110     faded = loop2.fade_in(fade_time).fade_out(fade_time)
111     print(time)
112     silent = AudioSegment.silent(duration=int(time)*2, frame_rate
      =44100)
113     fade_time2 = int(length * 2)
114     faded2 = loop2.fade_in(fade_time2).fade_out(fade_time2)
115     backwards = faded2.reverse()
116     velocidad_X = 1.001
117     so = faded2.speedup(velocidad_X, int(time), int(time)/2)
118     play(so)
119     play(silent)
120
121 def set_zero(sample, d):
122     argmax_idxs = sample.argmax(d)
123     return argmax_idxs*0.1
124
125 file_selected = '000000'
```

```
126
127 def accFeatures(data):
128     global model
129     global all_features
130     global file_selected
131     i = len(all_features)
132     if i == 20:
133         all_features = np.array(all_features)
134         features_scaled = min_max_scaler.fit_transform(all_features)
135         features_scaled = np.round(features_scaled, 1)
136         features_reshaped = np.reshape(features_scaled, (1,
    features_scaled.shape[0], features_scaled.shape[1]))
137         prediction = model.predict(features_reshaped, verbose=0)
138         prediction = set_zero(prediction, d=2)
139         print("Current Model Prediction: \n", prediction)
140         result = list(map(getClosestCandidate, prediction))
141         print("res:",result)
142         print ("File Selected from Dataset:", list(map(lambda x: x['
    file'], result)))
143         all_features = []
144         file_selected = list(map(lambda x: x['file'], result))
145         file_selected = str(file_selected)[1:-1]
146         file_selected = file_selected.replace("'", "")
147         playAudio("segments/segments_Movil2/" + file_selected,
    file_selected)
148         return file_selected
149     else:
150         all_features.append(data)
151
152 with sc.all_microphones(include_loopback=True)[-1].recorder(samplerate
    =sampleRate) as mic:
153   while True:
154       accFeatures(callback(mic.record(numframes=bufferSize).mean(axis
    =1)))
```

4.4. Conclusiones

Las redes LSTM secuencia a secuencia resultaron de gran ayuda para dotar a SEALI de los mecanismos necesarios que permitieran abstraer ciertas cualidades sonoras de la estructura de varias improvisaciones libres y emplear ese conocimiento para interactuar en tiempo–real consigo misma y con otros músicos, atendiendo a las oleadas energéticas y la actividad sonora producidas en una improvisación.

Los comportamientos de SEALI son de gran interés en términos de creatividad computacional ya que el modelo puede hacer una propuesta basada en su propio sesgo lo que promueve una exploración y derivas particulares de la improvisación en torno a los datos estructurales con los que fue entrenado. Además, si consideramos todos los cambios que ocurren en tiempo–real durante una improvisación; lo que suena en el paisaje acústico; el proceso de retroalimentación que involucra escuchar los propios resultados que genera SEALI; en combinación con las construcciones sonoras que el improvisador haga en un momento determinado; SEALI se ve expuesto a un entorno y espacio acústico siempre cambiante, homologado a la misma esencia de la transitoriedad que propone la improvisación libre. En este entorno resulta interesante preguntar, qué tanto SEALI conserva esa "huella digital" de la aproximación improvisativa de algún improvisador o improvisadores en particular en tanto que sus aproximaciones son completamente numéricas y no podemos hablar de una sensibilidad o escucha como tal, sino de una compleja relación numérica, en torno a las predicciones, y su comparación con una base de datos preexistente.

Estos resultados en un contexto social, médico o militar podrían significar un riesgo para tomar la decisión sobre si este modelo realmente podría ser útil o no, sin representar un peligro humano, ambiental o social. Un modelo con éste tipo de sesgo comprometería a quién tome la decisión de lanzarlo o no dado que podría representar un riesgo para las personas, animales, naturaleza o situaciones afectadas por el algoritmo. Sin embargo, en el campo del arte esto representa una valoración completamente distinta. Por un lado, como mencioné anteriormente, esto significa una forma en la que el modelo podría impregnar su algoritmicidad en la improvisación, lo cual desde mi perspectiva es algo deseable. A pesar de ello, esa algoritmicidad es siempre

determinista ante variables de entrada iguales, de manera tal que la exposición a variables de entrada siempre cambiantes posibilitan cierta flexibilidad y un sentido de novedad al sistema de interacción humano-máquina. Esta cualidad la entiendo como un espacio de posibilidad y exploración que no necesariamente podría ser leída como un error dentro del ámbito de la improvisación, sino incluso como un acierto, ese elemento aleatorio que puede ser tomado para conducir la improvisación hacia otro lugar, proponer materiales nuevos y mostrar algún grado de creatividad inherente al error, la falla, la falta de un nivel de precisión cien por ciento certero al momento de hacer predicciones, desde mi perspectiva, es lo que le da una mayor riqueza y posibilidades de proponer situaciones sonoras no previstas al algoritmo.

Capítulo 5

Bitácora de performances con SEALI

5.1. SEALI V.0.1

El caso particular de SEALI, permite profundizar en el entrelazamiento del ciclo práctica artística, desarrollo tecnológico e investigación e indagar en su dinamismo iterativo, así como las intra-acciones (Barad, 2007) de agentes humanos y más que humanos en contextos de improvisación libre.[1] En este caso particular, pensando a las intra-acciones como un espacio que integra la performatividad y dinamismo topológico[2] de agentes humanos y más que humanos en continuo surgimiento, comunicación y transformación entre sí, dentro de espacios agenciales sociales, artificiales y naturales.

SEALI en la versión 0.1 es un sistema interactivo de escucha automática capaz de identificar en tiempo–real cualidades y aspectos sonoros espectrales que incluyen al timbre, la cantidad de ruido o pureza espectral, la amplitud y el centro espec-

[1] En este sentido hablar de lo "no-humano", genera controversia al posicionar el término desde una perspectiva jerárquica antropocéntrica, así el término "más que humano" intenta desdibujar dichas jerarquías, y en su lugar pensar en un continuo fluir de afectaciones agenciales en constante performatividad. En este sentido, las fronteras entre animal y humano o máquina y organismo se vuelven ambiguas y, de este modo, se busca eliminar la noción del objeto o recurso fijo e inanimado que responde a la dicotomía esclavo-amo para pensar más bien en la intra-acción en actores o agentes.

[2] la propiedad de transitoriedad de los cuerpos para permanecer abiertos a la alteridad, el cambio y las transformaciones continuas

tral del sonido. A través de esta identificación, puede interactuar, dada una serie de instrucciones determinadas, con los elementos acústicos que escucha en torno a una improvisación. Cabe señalar que en esta versión aún no detecta elementos formales ni estructurales sobre la práctica de la improvisación libre, sino que requiere de un agente humano que controle algunos aspectos relacionados con la densidad y la cantidad de volumen general que se crea en el contexto de una improvisación, esto con el fin de acoplarse mejor en un contexto performativo a los improvisadores. El humano como agente externo para controlar estos aspectos de SEALI tiene la posibilidad de incidir como una especie de asistente que va guiando dichos elementos de la improvisación y puede activar o desactivar ciertos estados interactivos de SEALI, que incluyen; momentos contrastantes, caóticos, fluctuantes o que acompañan a un improvisador o a un ensamble de improvisadores. Para cada uno de los performances que presentaré en este capítulo, elaboré versiones específicas que responden de manera coherente con cada uno de los signos sonoros y lugares comunes a los que cada improvisador recurre. De este modo los materiales de cada improvisador fueron almacenados y analizados por SEALI para generar una identificación más certera en cuanto a los recursos particulares de cada improvisador.

Cabe destacar, que el desarrollo algorítmico de SEALI implica desde el comienzo un proceso recursivo ya que parte de la generación de una base de datos de materiales sonoros los cuales fueron generados por los improvisadores tomando como elementos detonadores la propia forma en la que el sistema "escucha". Esto lo hace desde las descripciones numéricas que diferentes descriptores de audio (*flatness, spectral centroid, MFCCs, loudness, spectral complexity,* etc.) posibilitan.

Una vez conformada esta información, el sistema la segmenta con base en cambios tímbricos sustanciales mediante el algoritmo SBic, tomando como compuerta los descriptores MFFCs. Estos segmentos son analizados por los descriptores de audio y posteriormente clasificados por un algoritmo de agrupamiento, finalmente esta nueva información es presentada a una red neuronal profunda la cual se encarga de generar un modelo que puede identificar las diferentes clases propuestas por el algoritmo de agrupamiento. El modelo puede ser ejecutado en tiempo real y hacer predicciones sobre nuevos materiales presentados.

Para la clasificación de los fenómenos sonoros y musicales, integré una tipología sobre los autómatas celulares propuesta por Christopher Langton y Stephen Wolfram (científicos de la computación y físico respectivamente) quienes introducen algunas reglas generales del comportamiento que pueden exhibir los autómatas celulares. Un autómata celular es un modelo matemático que evoluciona a pasos discretos dentro de una cuadrícula regular de celdas. Contiene un número finito de estados, el más básico sería encendido y apagado. La cuadrícula puede estar en cualquier número finito de dimensiones. Este tipo de modelos son adecuados para simular sistemas naturales en donde cada autómata celular puede ser descrito como una colección masiva de objetos simples que en interacción local con otros autómatas pueden exhibir resultados emergentes. Estos sistemas fueron descubiertos dentro del campo de la física computacional por John von Neumann a finales de la década de 1940 con su libro *Theory of Self-reproducing Automata.*

En el libro "A New Kind of Science" de Stephen Wolfram, plantea que la idea es explorar un nuevo universo abstracto: un universo computacional de programas simples que capturan la esencia de la complejidad y la belleza de muchos sistemas en la naturaleza. Introduce cuatro reglas generales que pueden presentar los autómatas celulares, estos pueden contener reglas de operación relativamente simples que en interacción con todos sus componentes internos su comportamiento general puede llegar a presentar una complejidad creciente que tiende a la generación de patrones regulares. Wolfram señala que estas cuatro reglas generales tienen el potencial de ser extrapoladas a fenómenos naturales en el entendido que la naturaleza presenta elementos simples que interconectados dan lugar a una serie de fenómenos emergentes complejos. Su tesis apunta a que los sistemas complejos en la naturaleza operan de forma similar a los autómatas celulares. Estas cuatro reglas generales de comportamiento de los autómatas celulares son presentadas de acuerdo con su creciente complejidad.

En la imagen 5.1 podemos observar estos comportamientos regulares que los sistemas de autómatas celulares son capaces de generar partiendo de distribuciones aleatorias de los diferentes agentes y funciones específicas que tienen que desarrollar. La idea de que existen 4 clases en los sistemas dinámicos viene originalmente del quí-

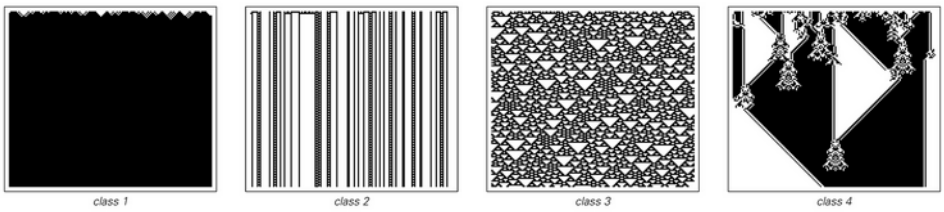

Figura 5.1: Comportamientos de los autómatas celulares.

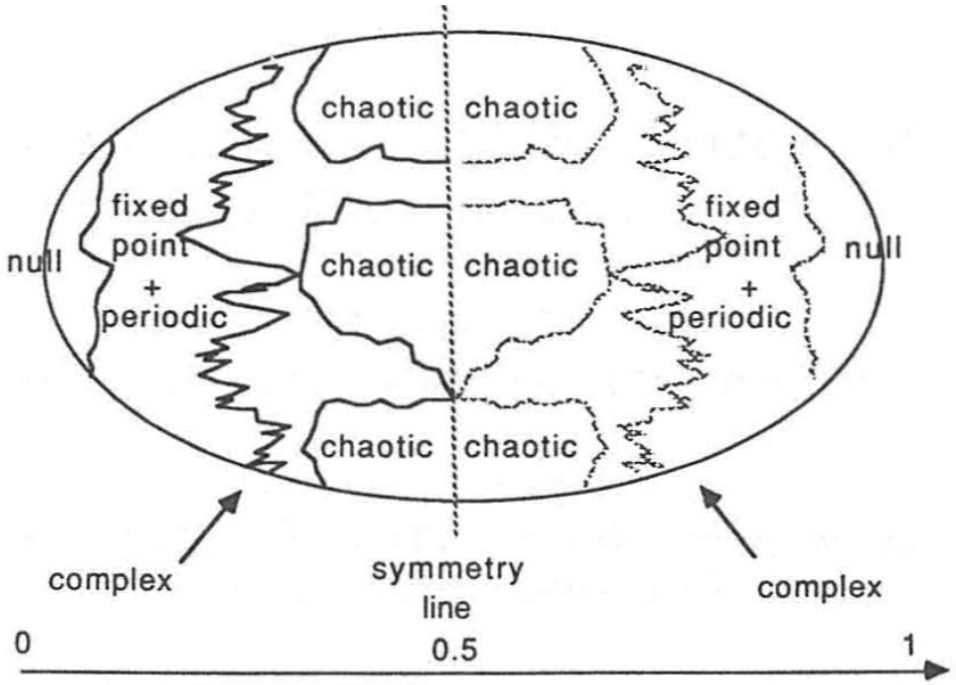

Figura 5.2: Comportamientos de los autómatas celulares.

mico Ilya Prigogine, quien identificó estas 4 clases en los sistemas termodinámicos: sistemas en equilibrio termodinámico (fijos), sistemas uniformes espacio/temporalmente (periódicos), sistemas caóticos y sistemas complejos lejos del equilibrio con estructuras disipativas.[3]

Wolfram señala que la clase 1 tiene un comportamiento muy simple y por lo general todas las condiciones iniciales llevan al sistema a un estado de uniformidad. Extrapolando esta idea a lo sonoro, encuentro esa uniformidad en los pedales (drones), secuencias sonoras que no cambian mucho en el tiempo y que al ser percibidas de manera holística dan un sentido de unidad. Un ejemplo de esta clase sería el ruido

[3] `https://www.wolframscience.com/nks/p240--four-classes-of-behavior/`

blanco producido por una computadora que aunque internamente está cambiando, dadas sus características pseudoaleatorias (números generados artificialmente al azar por un algoritmo), para la percepción auditiva se presenta como un fenómeno continuo y estable pese a los cambios discontinuos intrínsecos a su producción. Pero, si la velocidad en la que estos números son entregados fuera mucho menor, llegaríamos a percibir un fenómeno sonoro discontinuo. Otro ejemplo de un estado de uniformidad podría ser una onda sinusoidal la cual tiene una estructura periódica oscilante que se repite continuamente mediante la función seno y como fenómeno sonoro tiene un resultado continuo para la percepción.

La clase 2 para Wolfram podría presentar muchos estados finales diferentes, aunque todos consisten en ciertos conjuntos de estructuras simples que permanecen iguales o se repiten periódicamente. Trasladado a lo sonoro, serían sonidos que se repiten de manera intermitente, presentan una recurrencia constante, hay patrones claros, repetitivos y cíclicos. La periodicidad la podemos encontrar en la gestualidad, el timbre, la altura o la amplitud, es decir un sonido podría ser periódico aunque sus alturas cambien pero mantenga una gestualidad, un timbre o una amplitud cíclica. Todos éstos son elementos ayudan a distinguir la periodicidad de un fenómeno sonoro, así, lo que me interesa resaltar, es la reiteración gestual que permanece constante en el tiempo.

En la clase 3 “el comportamiento es más complicado, y parece aleatorio en muchos aspectos, sin embargo, triángulos y otras estructuras a pequeña escala se ven en algún nivel.” (Wolfram, 2002) Este tipo de fenómenos podrían parecer aleatorios para nuestra percepción pero en su forma no constituyen hechos propiamente aleatorios ya que son el resultado de múltiples dinámicas convergiendo simultaneamente. El caos se encuentra en sistemas dinámicos aperiódicos altamente sensibles a las condiciones iniciales los cuales son muy difíciles de predecir dada su naturaleza poco recurrente. A nivel musical encontraríamos diferentes estructuras sucediendo al mismo tiempo, además esta clase presenta cierta organización que no necesariamente es aleatoria sino difícil de asimilar. Así como en la clase 3 de la figura 5.1 podemos ver ciertas regularidades en las formas de los triángulos, en la música éstas regularidades es-

tán presentes en el tempo, la intensidad, la densidad sonora y en muchas ocasiones podemos encontrar comportamientos politímbricos y polimétricos.

Finalmente la clase 4, "presenta una mezcla entre orden y aleatoriedad: Se producen estructuras localizadas que, por sí mismas, son bastante simples, pero estas estructuras se mueven e interactúan entre sí de formas muy complicadas." (Wolfram, 2002) En el contexto musical el comportamiento de la clase compleja estaría en la frontera entre varias clases ya sea periódica y fija o caótica y fija o periódica y caótica, es decir, la presciencia de varios estados interactuando entre sí de manera simultanea. Este tipo de comportamiento puede ser visto en el cuarto recuadro de las imágenes antes mostradas donde podemos apreciar lineas rectas con cierta regularidad, patrones fijos, representados por el color negro y además estructuras sofisticadas que parecieran generar un orden de tipo fractal.

La potencialidad que encuentro al emplear estas categorías que en un principio podrían parecer muy generales es que me permiten aproximarme de forma sencilla a un problema que podría ser inmensamente complejo. Además una de las hipótesis iniciales fue que la improvisación libre puede ser definida a partir de estas cuatro categorías generales. La principal desventaja es que puede llegar a ser una aproximación muy determinista y tal vez forzada, además que presenta un sesgo o parcialidad por parte mía al momento de realizar la clasificación de los ejemplos. Por otro lado es posible encontrar ejemplos musicales que pertenezcan a varias categorías simultáneamente pero con distintos niveles de presencia, por ejemplo en donde ocurra que 70 por ciento pertenece a la clase fija y 30 por ciento a la clase periódica. Para este tipo de problema planteo hacer uso de clasificadores basados en lógica difusa que me permitan describir un ejemplo sonoro no de forma categórica sino de forma ambivalente.

A continuación muestro 4 ejemplos de las clases fija, periódica, caótica y compleja con 6 tipos de visualización. Estas visualizaciones son representaciones del espectrograma del contenido sonoro. La primera es un espectrograma con una frecuencia lineal, el segundo es el MFCC, el tercero es el espectrograma con la escala de Mel, el cuarto es el espectrograma con constante-Q, el quinto es el cromagrama, y el sexto es

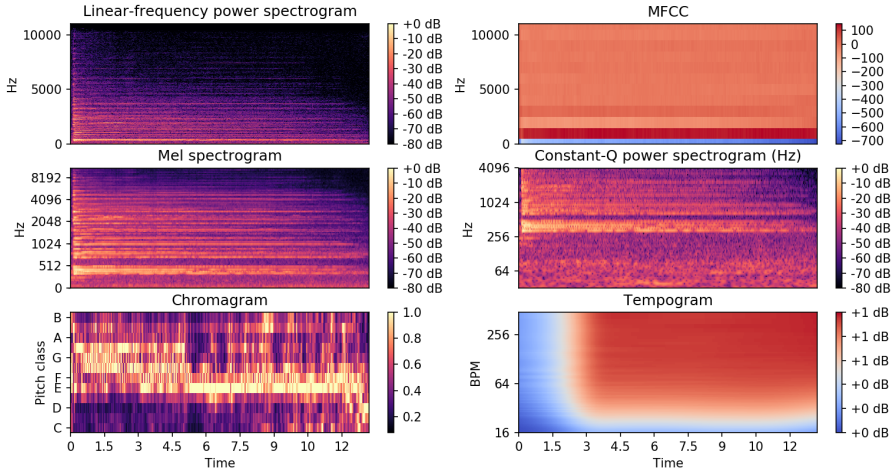

Figura 5.3: Clase Fija - Luigi Nono - Fragmente Stille An Diotima

un tempograma. Además de clasificar por medio de mi escucha, estas visualizaciones me han sido útiles para corroborar las clasificaciones ya que al visualizar el contenido espectral de cada sonido puede haber mayor claridad al diferenciar los ejemplos de audio. Por ejemplo, en la clase fija podemos observar claramente en el espectrograma lineal que hay líneas continuas sin ninguna interrupción lo cual indica un sonido estable con sus respectivos armónicos. Además, en el tempograma podemos apreciar a partir de la franja roja que no hay una distribución temporal única sino que es un bloque continuo que abarca un amplio rango de golpes por minuto, más o menos de 20 a 300 *bmp*, lo que indica que no hay una subdivisión rítmica única sino un amplio continuo rítmico, esto podría considerarse como un error pero en general es la forma en que el algoritmo del tempograma representa los sonidos fijos, así que podría decir que es consistente en su forma de representar esta clase y además esta información puede ser de gran ayuda para la clasificación.

Base de datos

La propuesta que planteé para la construcción de esta base de datos parte de la clasificación de los estados termodinámicos que propone Ilya Prigogine. Como he mencionado, estos pueden dividirse en cuatro estados; caótico, fijo, complejo y periódico. Cabe mencionar que estos estados no son exclusivos de ninguna música en

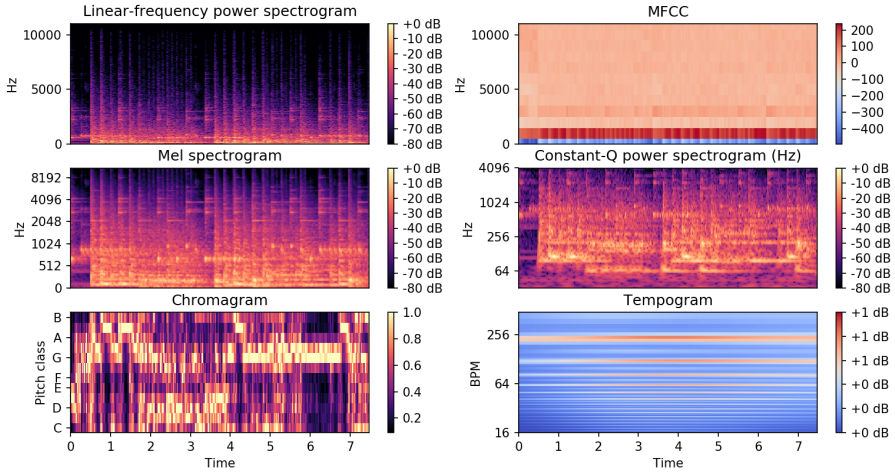

Figura 5.4: Clase Periodica - Henrry Cowell - Pulse

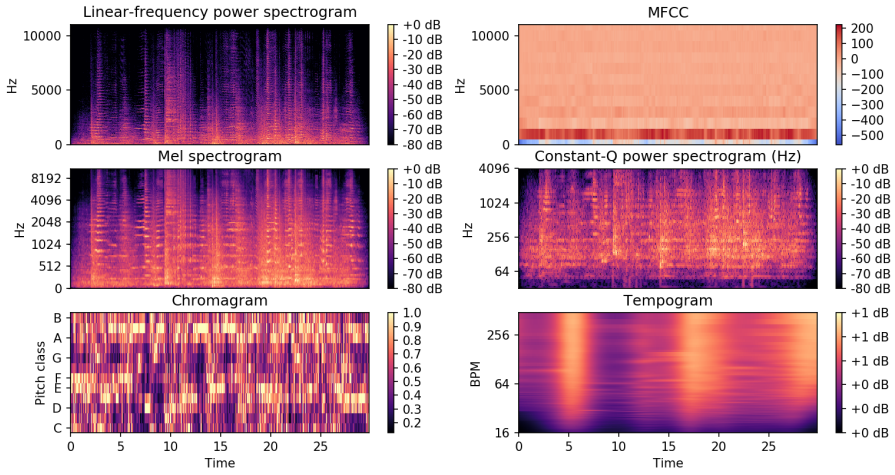

Figura 5.5: Clase Caotica - Iancu Dumitrescu - Nucleons Chaotiques/Transe

particular sino que busqué tendencias sonoras que pudieran ejemplificar de manera clara las definiciones propuestas por Prigogine y la reinterpretación de éstas enfocada en los comportamientos de los autómatas celulares definida por Stephen Wolfram en su libro “A New Kind of Science”. Realicé una revisión exhaustiva de escucha sobre muchos ejemplos musicales y sonoros para encontrar aquellos que se adaptaran a esta clasificación. Además “limpié” los archivos, es decir quité fragmentos que podrían causar discrepancias a la hora de entrenar el modelo. La base de datos está com-

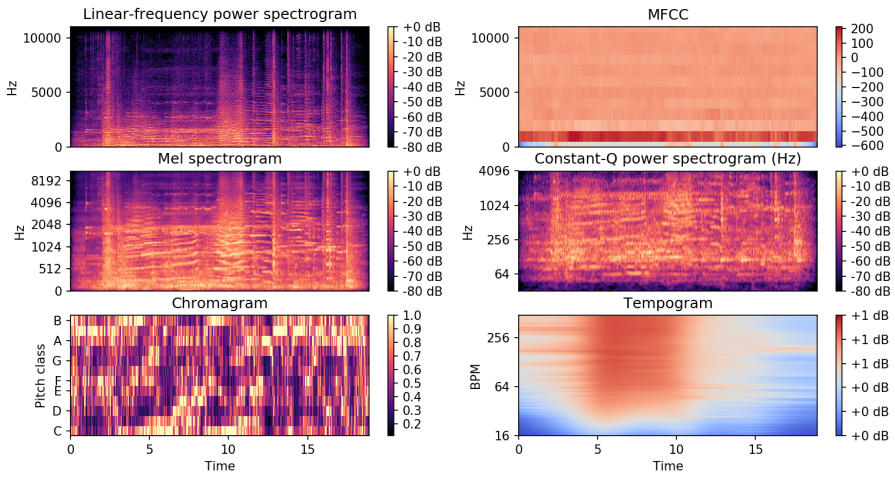

Figura 5.6: Clase Compleja - Iancu Dumitrescu - Nucleons Chaotiques/Transe

puesta por 495 archivos de audio con una duración total de 2.04 horas dividida en aproximadamente 30 minutos por cada categoría. En esta base de datos podemos encontrar paisajes sonoros compuestos por el repique de aves, el ululato de búhos, hasta motores de autos y lavadoras, por mencionar algunos ejemplos. Por otro lado hay gestos, frases y texturas de improvisaciones libres y composiciones que van desde ejemplos con una dinámica casi nula, pasando por pedales generados por platillos, hasta caudales de caos creados de manera sintética, superponiendo diferentes pistas musicales, improvisaciones y grabaciones personales. Así mismo, realicé una base de datos para pruebas compuesta por ejemplos diferentes pero similares a los que seleccioné en la base de datos inicial. Esta base de datos para pruebas contiene 23 minutos de audio divididos en 4 carpetas cada una con duración aproximada de 5.30 minutos de ejemplos por clase.

Para realizar la extracción de características de audio de las bases de datos trabajé con Supercollider y la librería SCMIR. Tomé la decisión de cortar en fragmentos de dos segundos cada uno de los archivos de audio y así obtener descripciones con una duración igual entre cada uno de ellos. Los descriptores utilizados fueron *chromagram, spectral flatness, spectral percentil y onset statistics*. La decisión por usar estos descriptores fue tomada a partir de los resultados obtenidos al experimentar con diversos descriptores, finalmente encontré que al conformar la base de datos con

estos cuatro descriptores podía obtener una mejor clasificación la cual será descrita más adelante. El código desarrollado en Supercollider tiene la opción de generar descripciones del audio de los ejemplos proporcionados aunque sean de diversos tamaños. Una desventaja de esta implementación es que si uno de los archivos tiene una duración menor al tamaño de la ventana de corte será inmediatamente ignorado, por lo que no será tomado en cuenta como un ejemplo y será descartado.

Por otro lado, trabajé con las librerías de Tensorflow y Keras para la generación de un modelo predictivo basado en el entrenamiento recursivo de la base de datos inicial. Para esto empleé algoritmos de redes neuronales profundas y redes neuronales convolucionales. La aproximación para realizar este tipo de entrenamiento es el aprendizaje supervisado, que requiere de una clasificación anotada misma que realicé al clasificar los archivos de audio en carpetas diferentes. Comencé por experimentar con distintas configuraciones y atributos de las redes neuronales para intentar generar un modelo de predicción que fuera lo suficientemente robusto para predecir a qué clase pertenecen los nuevos ejemplos mostrados.

Adicionalmente, trabajé en la parte de comunicación entre las distintas aplicaciones utilizadas, para este proceso primero inicio el servidor de audio de Supercollider (scsynth) que se encarga de activar una entrada de audio, ya sea interno o externo, para grabar archivos de audio de dos segundos de duración. Después, el extractor de características de audio, se encarga de extraer los datos de los archivos de audio generados mediante la librería SCMIR. Aquí comienza la intercomunicación entre:

• El servidor OSC de Supercollider encargado de la comunicación con Python OSC server; envía los datos extraídos y recibe nuevamente los datos que Python OSC server regresa.

• El servidor OSC de Python se encarga de recibir los datos de Supercollider y convertirlos al formato JSON para comunicarse con Tensorflow model server.

• Tensorflow model server recibe los datos en formato JSON desde Python y los compara con su modelo almacenado, nuevamente envía de vuelta el resultado de la

clasificación sugerida por el modelo a Python y esta respuesta es enviada, a su vez, a Supercollider mediante el protocolo OSC.

Por otro lado, la parte reactiva del sistema utiliza procesos de sonificación y lógica booleana que activa distintos tipos sintetizadores de acuerdo con la clasificación generada por el modelo: por ahora está compuesta de síntesis sonora y el procesamiento de los archivos de audio tomados de la base de datos anotada.

Sonidos sintéticos:

• Sinusoidales puras controladas por procesos de sonificación de los datos de entrada de cada una de las clases.
• Sinusoidales puras controladas por la detección en tiempo real de la frecuencia (a través de escucha de máquina).
• Sinusoidales procesadas mediante síntesis granular.
• Síntesis sustractiva de ruido blanco y rosa con filtros resonadores controlados por la detección en tiempo real de amplitud, ataques y frecuencia.
• Síntesis sustractiva de ruido blanco y rosa con filtros resonadores controlados por los datos de entrada de las predicciones de cada una de las clases.
• Generador de síntesis estocástica dinámica concebida por Iannis Xanakis (Gendy)(Serra, 1993), (Xenakis, 1992).

Archivos de audio:

- Lanzador aleatorio de archivos de audio que también varía aleatoriamente la velocidad de reproducción de cada archivo de la base de datos.

- Lanzador aleatorio de archivos de audio procesados por síntesis granular controlada mediante los datos de entrada de cada una de las clases.

La forma de reacción/interacción general en esta fase de desarrollo de SEALI fue activar estados contrarios que permitieran mantener una forma de interacción dinámica un su conjunto, con esto traté de generar una especie de equilibrio entre los distintos estados. Además, esta primer aproximación fue dividida por rangos

mapeados de acuerdo con los datos recibidos, de manera que si el sistema detecta algo:

- **Fijo:** su reacción está delimitada por sonidos de la base de datos caóticos y sonidos complejos producidos por la síntesis sustractiva de ruido blanco, controlados por la detección en tiempo real de amplitud, ataques y frecuencia.

- **Periódico:** está dividido en dos partes, una parte activa sonidos de la base de datos periódica, y la otra transformaciones granulares sobre sonidos de la base de datos caótica.

- **Caótico:** en general su reacción está delimitada por elementos sonoros que tienden hacía lo fijo y al silencio.

- **Complejo:** está dividido en tres partes, una parte activa sonidos de la base de datos compleja, la segunda, sonidos fijos y la tercera sonidos sintéticos que tienden a lo complejo mediante síntesis granular.

Asimismo, desarrollé distintas formas de interacción que son activadas al identificar, a partir de la escucha de la forma general de una improvisación, los modos de interacción que he detectado en mi práctica y dentro de las improvisaciones colectivas, estos son: escucha, imitación, proposición (de una nueva idea musical), acompañamiento, ruptura y solo. Al escoger alguno de estos estados SEALI, podría retar, acompañar, proponer o imponer(se) sonoramente frente a un humano, y, generar una dinámica entre los diferentes modos de interacción posibles.

Cabe mencionar que a lo largo del desarrollo de esta versión me he percatado que, pese a las múltiples aproximaciones, técnicas y ajustes empleados para la resolución de esta tarea, es muy complicado obtener un buen modelo de aprendizaje de los elementos sonoros que estoy esperando categorizar. Además, pese a haber limpiado detenidamente la base de datos y probar su rendimiento con distintas arquitecturas de redes neuronales los resultados en cuanto a la clasificación siguen siendo deficientes. La principal causa la atribuyo a que la base de datos no es lo suficientemente representativa (amplia) como para que el modelo pueda generalizar los datos presentados y hacer clasificaciones más acertadas en la práctica de la improvisación libre.

Pese a esta y otras limitaciones he probado el sistema en funcionamiento con diversos improvisadores realizando algunos ejercicios para corroborar su performance. Uno de estos ejercicios consistió en que los improvisadores tratáramos de generar los componentes sonoros adecuados para que el sistema permaneciera en una sola clase durante cierto tiempo. ¿Qué es lo que el sistema necesitaba escuchar para devolver alguna de las clases que estaba entrenado para detectar? Lo que encontramos aquí fue una tendencia en SEALI donde generó transiciones entre alguna de las clases y la clase compleja, por ejemplo, entre fija y compleja o periódica y compleja. La clase caótica resultó difícil de generar debido a la intrincada relación entre los sonidos superpuestos de los ejemplos que contiene esta clase. Solo logramos activarla en algunos momentos pero inmediatamente después predominaba la clase compleja. Además probamos que hay una clase indeterminada que surge de la combinación entre cada una de las clases, a este elemento lo llamé clase indefinida, por el momento, pero mi suposición es que pertenece a la clase compleja ya que por definición lo complejo tiene un poco de cada una de las otras clases. Nuevamente, hizo falta una base de datos mucho más amplia para poder generalizar los cuatro tipos de clases propuestas al modelo.

A continuación presento algunas exploraciones creativas de los performances derivados de esta aproximación que tomaron en cuenta el reajuste de la base de datos.

5.2. Sesiones de improvisación Resistencias maquínicas con SEALI V.0.1

El concepto de resistencia que da nombre a este ciclo de conciertos es muy amplio y habría que especificar desde dónde se entiende y cuál es esa resistencia maquínica. Una interpretación de resistencia viene desde la electrónica, en este contexto implica la oposición que presenta un material al ser atravesado por una corriente eléctrica; en otras palabras una fuerza que deja pasar cierta parte de la corriente eléctrica, la transforma y hace que su flujo energético cambie, el cual generalmente se reduce. Por otro lado, el diodo sería el componente que cierra y no deja pasar nada fuera del circuito que crea, es como una compuerta lógica binaria y determinista. Desde otra mirada, ¿cómo entendemos la resistencia desde una perspectiva socio-política y cómo

ha evolucionado a través de las diferentes épocas y circunstancias sociales?, ¿cómo la entendemos pensando en las resistencias zapatistas por ejemplo? Pensando en que la resistencia podría manifestarse de forma violenta o pacífica y de diversas formas, en un principio, no sería algo que permanece estático en el tiempo, sino más bien en una constante performatividad transitoria y adaptativa. En esa multiplicidad de posibilidades performáticas que toma desde las resistencias sociales, es una fuerza que se opone a los modelos opresores instaurados por el mal gobierno y las malas administraciones. Las prácticas de resistencia social son una alternativa a la dureza que presentan las estructuras hegemónicas instauradas. Sin embargo, en este momento donde vemos crecer un capitalismo salvaje, *hardcore*, adaptativo y resiliente, cabría preguntarse si la resistencia sigue siendo una alternativa (tal vez bastante romantizada) contraria a dicha estructura que se caracteriza por llevar a cabo prácticas de mutilación, explotación y escasez. Así, entiendo a la resistencia como algo que va más allá de estas fuerzas contrarias, ya que ambas integran propiedades resilientes y resistentes, adaptando sus mecanismos homeostáticos en un medio cambiante que les permiten replantear sus modos de operación ya sea para luchar por mejores condiciones de vida o de explotación. En este escenario ¿dónde queda la resistencia como una alternativa otra cuando lo que creíamos duro se vuelve también resistente? La resistencia, entendida en términos de adaptabilidad, resiliencia, abierta a la transformación y el cambio es también una fuerza que emplean las estructuras hegemónicas. En ese sentido, toda estructura resiste aquellas fuerzas contrarias, ablandando su propia dureza, abriéndose y dejando pasar así las corrientes para adaptarse con mayor aptitud al cambio que finalmente determina su núcleo, permitiéndole habitar la desterritorialización de su dureza y transitar por esos territorios blandos, abiertos, transmutables.

La resistencia maquínica la encuentro en ese espacio de cuestionamiento, ruptura y reivindicación constante de los automatismos presentes en mi propia práctica musical-compositiva-creativa-improvisativa. Ello despierta la posibilidad de que un sistema como SEALI, concebido como un mecanismo automatizado pueda servir de ayuda para romper ciertas tendencias automatizadas al hacer música. Cómo, SEALI es capaz de cuestionar esos hábitos a partir de sus propios automatismos que parten

de perspectivas de intra-acción no lineales e indeterminadas donde, en algunos casos, los modelos entrenados en el estilo de un improvisador sumamente experimentado podrían empujar la improvisación hacia lugares aún no explorados. Esto, lo entiendo como una forma de resistencia donde ciertamente hay una serie de reglas fijas y estructuradas que responden de forma automatizada pero a través de una dinámica intra-activa que evoluciona y sigue transformando los espacios de posibilidad creativa. La posibilidad de tener múltiples base de datos funcionando simultáneamente y que son empleadas como la base de la intra-acción producida por SEALI puede ser un punto de partida pero no solo se queda ahí, cómo veremos en el performance de resistencias máquinas I con Jorge Berumen y Diego Villaseñor. En éste, es posible escuchar que los cambios tímbricos y sonoros que el sistema plantea, dirigen a la improvisación hacia lugares que no estaban previstos por los músicos, empujando cada vez más la improvisación hacia un lugar que ninguno de los agentes implicados tenía previsto. Una forma de resistencia hacia esos automatismos ligados a la propia programación de SEALI. Entonces los elementos imprevisibles y azarosos de la máquina, dentro de su espacio de posibilidades y por su carácter tendente al caos, pueden dirigir a los improvisadores a lugares de encuentro inesperados. Luego, otras ideas que resuenan como parte de la conceptualización de este ciclo de conciertos son las resistencias que puso el propio proceso de investigación para entender cabalmente cómo el desarrollo tecnológico y performático podrían comunicarse para ser llevados a la práctica. Asimismo, la elección metodológica que se resiste a ser descubierta, a funcionar, que propone una serie de caminos posibles y que incluso al finalizar esta tesis siguen abiertos. Desde estas reflexiones es que surge la idea de la resistencia como una fuerza que se opone y cuestiona pero que al mismo tiempo deja pasar ciertas cualidades, conocimientos y certezas, para aprender de esa cualidad transformativa que la intra-acción humano-máquina propone.

5.2.1. Blackbox: Jorge Berumen y Diego Villaseñor

La primera muestra que realicé presentando los primeros avances de SEALI fue un performance en vivo en el que participaron Diego Villaseñor en la guitarra eléctrica, Jorge Berumen en la batería y yo controlando algunos parámetros en tiempo real y activación de los modos de reacción con SEALI. Esta presentación tuvo lugar en el

foro BlackBox del Centro Nacional de las Artes dentro del marco del programa ACT (arte ciencia y tecnología) UNAM. Lo que ocurrió ahí en cuanto a mi interacción, lo entiendo como una extensión sensible e inteligente que se agregó a SEALI, una conjunción colaborativa humano-máquina, ya que mi intuición y conocimiento de forma musical e improvisación libre se sumaron a las propuestas sonoras de SEALI. Esta performance fue una buena experiencia para probar y entender las limitaciones y potencias que hasta ese momento, tuvo SEALI. Al finalizar la intervención sonora se abrió la conversación con el público asistente quienes mostraron un gran interés por el proyecto y externaron sus impresiones respecto a este tipo de tecnologías en el marco de la creación musical.

Por su parte, los improvisadores invitados tuvieron impresiones diferentes; Berumen mencionó que no sentía que hubiera una respuesta en tiempo real por parte del sistema y que eso lo desconcertó, pero que en el constante fluir musical sí sentía una interacción más acertada. Villaseñor mencionó que el sistema realmente lo estaba acompañando y siguiendo, por momentos retando a cambiar sus materiales. Mencionó que hubo una amalgama en la interacción, el sistema se acopló a lo que estaba escuchando y a lo nuevo que se estaba construyendo de manera colectiva. A partir de esta presentación surgió un registro sonoro y visual que forma parte de la documentación de la investigación. A continuación comparto los enlaces al registro de esta performance:

`https://www.youtube.com/watch?v=fFpvOIXpypY`

5.2.2. Bucareli 69: Sarmen Almond y Diego Villaseñor

Para esta versión realicé una pequeña serie de instrucciones que me permitieron trabajar una versión donde SEALI detectaba entre sonoridades producidas con la voz y la guitarra eléctrica los diferentes materiales que ambos improvisadores proponían al sistema, basándome en 4 categorías que fueron recopiladas a través de la base de datos hecha a mano, la cual contemplaba sonidos fijos, periódicos, caóticos y complejos. De manera tal que, al reconocer ciertos elementos sonoros el sistema respondiera con elementos contrarios a lo que reconocía en un momento determinado; si escucha algo

fijo responde con un material periódico, periódico a fijo, caótico a complejo y complejo a caótico.

En esta versión algo que busqué fue generar un diálogo entre elementos fijos y móviles dentro de un contexto *comprovisacional* (Dudas, 2010), ya que parte de la base de datos utilizada para generar esta versión comprendió algunos de los materiales sonoros de *tombstones* de Michael Pisaro, de manera que, la música tocada por los improvisadores generará un diálogo con esta composición.

La interacción en este performance fue muy interesante ya que al explorar activamente cómo los materiales que ellos producían iban, de algún modo, activando los elementos sonoros con los que el sistema respondía, esto les permitió pensarse, además de improvisadores, como activadores y detonadores de una serie de procesos, sonidos y bancos de sintetizadores que estaban programados para surgir cada que los improvisadores emitían sonido. En esta comprovisación fue interesante ver cómo los elementos sonoros que ellos proponían y los elementos sonoros que el sistema regresaba, iban surgiendo y desapareciendo, llevando a los improvisadores a un espacio de exploración sonora muy específico, que desde mi parecer, acotó los lineamientos estéticos a los que ellos mismos llegaron. Otro elemento que ocurrió en el performance fue que entraron en un bucle junto con la máquina que ciertamente seguía reaccionando de forma similar en ese momento ya que, no solo ellos, sino todos los agentes implicados en ese sistema entraron en esa especie de bucle que los llevó a interactuar desde la propia algoritmicidad de los códigos, la incidencia acústica entre los improvisadores, solos y en interacción con SEALI, y la incidencia que SEALI tenía sobre ellos. De esa manera fue como empezaron a entablar un diálogo con SEALI que produjo una serie de afectaciones que atravesaron a todos los escuchas que presenciaron ese momento de creación sonora. A continuación comparto el audio de este registro:

`https://archive.org/details/villasenor-almond`

5.2.3. Bucareli 69: Jerónimo García, Fabian Rangel, Diego Villaseñor y Sarmen Almond

Para esta versión realicé una versión de SEALI que reaccionara a cuatro elementos sonoros, cressendos, sonidos largos, ataques y trémolos, para ello realicé una base de datos que contuviera muestras representativas de cada una de estas clases; de manera que SEALI pudiera adaptarse a la amplia gama de sonidos que le presentaran los improvisadores. Esta diversidad contempló un piano extendido suspendido en el aire creado por Jerónimo García, el cual utiliza cuerdas de piano tensadas de distintos calibres para excitar diversas membranas de parches de bombo. Las cuerdas pueden ser activadas mediante un teclado al cual están conectadas, también pueden ser frotadas, golpeadas, jaladas, pellizcadas, percutidas, raspadas, etc. con diversos tipos de "exitadores". Además en esta improvisación había violín tocado por Fabián Rangel, Diego Villaseñor tocó la guitarra eléctrica y Sarmen Almond se integró a la improvisación con su voz. A continuación un extracto del performace:

`https://www.youtube.com/watch?v=K6XwoiKOXOg`

5.2.4. Hangar, TopLap Barcelona (en línea): Aarón Escobar Live Coding/SEALI Retroalimentado

A través de este performance, en esta propuesta busqué generar narrativas transmediales que permitieran despertar reflexiones y cuestionamientos sobre el significado del proceso de acelareación en la comunicación digital que atravesamos al inicio de la pandemia en marzo de 2020, no desde una convicción consensuada colectivamente, sino desde una emergencia coaccionada por las circunstancias globales de ese momento. Algunos de los cuestionamientos y planteamientos que estuvieron presentes para realizar este performance derivaron de las indagaciones sobre la tecnología y su afectación en el ámbito social de (Braunstein, 2011):

> "¿Cuáles son los límites entre lo digital y lo viviente? ¿Hasta dónde llega el dispositivo, somos nosotros el dispositivo que alimenta a los dispositivos? ¿Podemos distinguir entre lo vivo y sus prolongaciones protésicas en un momento donde estamos regulados por sustancias y artefactos inventados

> por la técnica? ¿Cuáles son las alternativas cuando cada vez hay más cyborgs, organismos cibernéticos que surgen entre el cruce de lo orgánico y los mecanismos que regulan su funcionamiento?" (Braunstein, 2011)

Además, los procesos de adaptación que tenemos los organismos vivos del planeta pareciera que cada vez son más veloces ante las injurias de la actividad humana en el planeta producto del desarrollo del sistema social dominante. Como especie humana esto no podría ser diferente, ante una situación como la actual, donde los virus pueden causar una inestabilidad social e incluso vulnerar nuestra salud, podemos exhibir de manera muy particular patrones resilientes. Nuestra evolución nos ha llevado a desarrollar estos mecanismos homeostáticos que parecieran ser suficientes para contrarrestar cualquier amenaza que transgreda nuestra supervivencia. Nuestras prótesis tecnológicas, desde hace mucho tiempo atrás, dejan ver que nuestra naturaleza no se limita a las posibilidades físicas, biológicas, psíquicas o sociales sino que se extienden de múltiples formas hacia caminos que aún no conocemos, tal como la transición irremediable a un mundo donde la actividad digital humana se vuelve eminentemente necesaria.

`https://www.youtube.com/watch?v=59DVGLfFmZE`

5.2.5. Fam UNAM: Aarón Escobar Guitarra eléctrica + SEALI

Otro performance que realicé junto a SEALI fue para el coloquio de alumnos del posgrado de la UNAM en la facultad de música. En este a través de una improvisación con guitarra eléctrica y algunos objetos y silbatos de viento, interactúe con SEALI reaccionando a los elementos fijo, caótico, periódico y complejo. Aquí el enlace a esta impovisación.

`https://www.youtube.com/watch?v=ERfJlKyQ_wo&t=11s`

5.2.6. Manuel Enríquez Movil II

Durante este recorrido, trabajé la pieza *Movil II* del compositor Manuel Enríquez , junto con el violinista e improvisador Fabián Rangel. Para esta versión grabamos individualmente cada uno de los gestos de la partitura abierta de Enríquez, lo cual

nos permitió entrenar el modelo computacional de SEALI con estos materiales, de manera que al ser escuchados pudiera reaccionar consecuentemente a lo largo de la pieza. Para entrenar al modelo decidimos reducir todos los materiales de la partitura a cuatro elementos principales que engloban, grosso modo, el cuerpo de sonoridades más contrastantes de la pieza. Ello nos condujo a organizar el proceso de grabación en cuatro clases: tremolando, crescendo, sonidos largos y ataques. Así, dentro de estas "técnicas generales" quedó englobado el matiz de sonoridades que el modelo computacional podía identificar al momento. Cabe destacar, que estas grabaciones fueron transpuestas a varios tonos para tener una base de datos de entrenamiento más sustancial y representativa de cada una de las técnicas, logrando una totalidad de 500 muestras por cada una de las clases y un total de 1.6 GB. Con el ánimo de invitar a la exploración y experimentación, a continuación presento la base de datos supervisada que contiene las grabaciones y transposiciones utilizadas para entrenar el modelo con redes neuronales profundas.

`https://archive.org/details/materiales_movil_II`

El proceso de montaje de esta obra fue muy enriquecedor debido a que la apertura de la obra de Enríquez, permite explorar espacios de creatividad musical donde la composición y la improvisación se difuminan, posibilitando la exploración abierta hacia múltiples modalidades en las cuales la obra puede ser abordada. Enríquez especifica que la aproximación gestual de la obra puede ser "al azar", tocarse con el "máximo de fantasía" y pudiéndose aproximar a ella de tres formas: acústica, amplificada o con la una cinta pregrabada. Al no tener acceso a ésta, tomamos la decisión de incluir a SEALI en lugar de la cinta pregrabada, integrándose desde la síntesis sonora, el procesamiento de la señal del violín en tiempo–real e incidiendo esporádicamente con algunos de los fragmentos de la base de datos grabados de acuerdo con la predicción del modelo de clasificación.

La programación de la respuesta a las predicciones del modelo, parte de una escucha atenta al discurso musical, en donde la interpretación de la obra por parte del violinista me condujo a buscar una gestualidad en SEALI que le permitiera, más que generar un resultado sonoro homologado a nivel tímbrico, entablar una forma

de interacción equilibrada, mediante la introducción de cierto contraste e imitación a lo que Fabian planteaba. Esto fue posible al establecer reglas de reacción limitadas por compuertas y en algunos casos contrarias: si SEALI escucha sonidos largos responde con ataques y viceversa; si escucha crescendo responde con tremolando, etc. Aunado a ello, SEALI escucha el violín directamente por un micrófono piezoeléctrico (aislado del entorno sonoro), puede escuchar el propio resultado sonoro que genera mediante un micrófono ambiental o también escuchar de manera aislada su propia salida[4]. La combinatoria que estas tres modalidades posibilitan, o bien, puede producir un resultado muy claro de identificación gestual –al escuchar solamente al violín aislado del contexto–, o puede ocasionar que sus reglas, hasta cierto punto, entren en conflicto ya que si escucha un sonido largo, a lo cual responde con ataques, entonces el resultado sonoro es un sonido complejo que incluye a los sonidos largos y los ataques, lo cual no estaba previsto en el modelo de escucha de SEALI. Desde esta perspectiva el sesgo de la escucha introducida e inducida es lo que produce un resultado emergente que para mi escucha resulta sumamente interesante ya que las interacciones entre el violín y la electrónica pueden ser muy variadas; por momentos acompañándose, en otros rompiendo con el esquema y proponiendo nuevos elementos al discurso comprovisativo.

Una autocrítica que hago a esta implementación temprana de SEALI, es que pude haber trabajado un modelo que integrara sonoridades complejas, producto de la retroalimentación entre la salida de la electrónica y la gestualidad sonora del improvisador, para tener mayor control sobre las decisiones que SEALI toma al momento de escuchar esta conjunción. Además, si se quisiera evitar la retroalimentación se podrían correr dos modelos simultáneamente, el primero escucharía únicamente las técnicas de violín con el piezoeléctrico y el otro la salida de SEALI pasando internamente por jack. Sin embargo, este es uno de los muchos aspectos que quedan por cubrir en esta primera versión.

[4]Al enviar la salida de Supercollider a la entrada del módulo de escucha de essentia y py audio mediante una conexión interna vía jack audio connection

A continuación presento el resultado obtenido de la performace de esta obra suscitada en el marco de la "tocada del miércoles" organizada por música UNAM y grabada en el estudio de Pablo García Valenzuela (Pablo Gav).

`https://youtu.be/fVrCt4X1Y7E?t=731`

5.2.7. Interdictos protésicos

El desarrollo algorítmico de SEALI en la versión programada para la comprovisación(Dudas, 2010) *Interdictos Prostéticos*[5] implicó desde el comienzo un entrelazamiento creativo humano-algorítmico. Esta versión parte de la generación de una base de datos de gestos sonoros los cuales fueron grabados por los improvisadores tomando como elementos detonadores ciertas características del proceso de escucha automática de SEALI. Como he mencionado en capítulos previos, la escucha automática involucra el proceso de extracción e identificación de descriptores de audio (algunos de éstos son: *flatness, spectral centroid, MFCCs, loudness, spectral complexity, onsets*) (Vinet *et al.*, 2002). Partir de un elemento ligado al proceso en que SEALI escucha para generar gestos sonoros, busca integrar un continuo orgánico-digital donde la performatividad sonora del músico y la aproximación hacia la detección de elementos sonoros están entrelazadas. Para ello, trabajé individualmente con seis intérpretes (Diego Villaseñor en la guitarra eléctrica, Fabián Rangel en el violín, Gilberto Ramón Celis en el contrabajo, Elena Martínez en las percusiones, Diego Villa en el piano y Aida Contreras en la flauta transversa). El proceso de colaboración fue muy fructífero ya que los instrumentistas se mostraron muy interesados e interpelados con la pieza y la propuesta de colaborar en ella, no solo como intérpretes de la obra, sino como agentes que influyen y construyen directamente el sistema interactivo con su propia exploración musical.

En estas sesiones de trabajo les propuse a los intérpretes generar materiales a partir del concepto de escala tomándolo como detonador de diferentes aproximaciones a las cualidades acústicas y prácticas de su instrumento. Cabe mencionar que delimité estas exploraciones a tres parámetros: aproximaciones a la digitación, envolventes acústicas y contenido espectral que puede dividirse en brillo, complejidad, densidad y

[5] `https://archive.org/details/aaron-escobar-interdictos-protesicos`

centralidad espectral. Estos materiales fueron analizados numéricamente para crear una base de datos a través de descriptores de audio que representan el enfoque con el cual el sistema "escucha" y fue entrenado. Por ello consideré importante apoyarme de la forma en que el sistema reconoce materiales musicales y extrapolarlo a las exploraciones de los músicos, de manera tal que hubiera una sinergia en cuanto a lo que los músicos proponen desde un inicio y las cualidades de reconocimiento que tiene el sistema.

Una vez recopiladas estas grabaciones fueron segmentadas con base en umbrales tímbricos sustanciales, estos segmentos se analizaron por los descriptores de audio y posteriormente fueron agrupados por similitud tímbrica mediante el algoritmo *Mean Shift Clustering* (Comaniciu y Meer, 2002), finalmente esta reagrupación de los datos fue analizada mediante una red neuronal profunda que generó un modelo capaz de identificar las diferentes clases propuestas por el algoritmo de agrupamiento. El modelo puede ser ejecutado en tiempo-real y hacer predicciones sobre nuevos segmentos presentados. Para el conocimiento estructural de las improvisaciones recopilé una base de datos con improvisaciones libres la cual fue segmentada y analizada con la misma aproximación anterior, generando así una base de datos supervisada. Esta información es reorganizada en series de tiempo y analizada por una red neuronal recurrente (Long-Short Term Memory) (Hochreiter y Schmidhuber, 1997) la cual es capaz de predecir secuencias temporales a corto, mediano y largo plazo. A través de estos procesos de reconocimiento, el sistema interactivo en tiempo–real de SEALI va tomando decisiones a nivel tímbrico, energético e interactivo de manera que propone gestos sonoros similares o contrastantes a los que está escuchando de acuerdo con el conocimiento que tiene sobre lo que ocurre en el contexto de una improvisación. A continuación presento la versión ensamblada que realicé para la concreción de esta obra:

`https://bit.ly/3qagvMh`

Asimismo, esta obra permite su interpretación en diferentes modalidades, con distintas instrumentaciones e intérpretes. En el performance de *Interdictos Prostéticos*[6] que realizó Miguel Ángel Cuevas, ocurren varios fenómenos de retroalimentación

[6]Interdictos Prostéticos versión de Miguel Ángel Cuevas `https://youtu.be/OUjAYWYenco`.

que interpelan desde la adecuación de los diferentes elementos tanto al improvisador como a SEALI, si se está escuchando a sí mismo. Es importante resaltar que en esta performance la intra-acción de ambos agentes no está fija ni estructurada en el tiempo, sino que continuamente se produce abriendo espacio a la emergencia en términos de contingencia, azar e indeterminación. Uno de estos fenómenos, es la propuesta musical estructural que hace el sistema basándose en el estilo particular que algún improvisador propuso en un álbum. De esta manera se conserva cierta esencia a nivel estructural con la cual el sistema transita la improvisación desde ese conocimiento que le permite adecuar su comportamiento dadas ciertas situaciones sonoras. Este proceso podría ser visto como un agente que abstrae el estilo musical y trasciende el espacio-tiempo-materia insertándose en ese continuo e indivisible fluir iterativo en contextos completamente distintos. Al conservar ciertos rasgos cadenciales identificables por un improvisador, la improvisación puede conducirse en términos estructurales desde ese gran conocimiento que caracteriza este estilo.

5.3. Experiencia en el FIMNME con SEALI V.0.2

Antes de poder finalizar esta tesis, tuve la oportunidad de presentar a SEALI en interacción con con la obra abierta Interdictos Protésicos y el Movil II de Manuel Enríquez en el Foro Internacional de Música Nueva Manuel Enríquez.[7]

5.3.1. Interdictos protésicos con SEALI V.0.2

Para la primera obra, tuve la posibilidad de trabajar con 6 músicos (Mariana Chávez, Carla Benítez, Carlos Rangel, Noé Macías, Edwin Tovar y Vladimir Ibarra) y un director musical (Rodrigo Cadet) los cuales experimentaron sobre las posibilidades abiertas que la obra plantea e interactuando con SEALI durante 5 días de ensayo antes de su estreno.

Durante estos ensayos, fue sumamente interesante escuchar que, en algunos casos, las exploraciones sonoras de los músicos estuvieron condicionadas a homologarse a los timbres que SEALI les proponía. Para esta versión decidí dirigir la reactividad de SEALI hacia exploraciones con sonoridades largas resonantes y brillantes, con

[7] `https://issuu.com/cnmo.inba/docs/programa_2022_octubre-email`

una fuerte inclinación a la superposición de múltiples sinusoidales, la imposición de sonoridades tendentes al ruido y el contraste con el silencio que SEALI proponía al responder a los lineamientos de la forma musical de la improvisación libre de Hans Reichel; en particular a sus grabaciones para el disco Coco Bolo Nights. En este sentido la aproximación de SEALI a nivel sonoro respondía de maneras en las que Reichel jamás lo hubiera hecho, pero basándose en el análisis que el modelo LSTM hizo a los elementos formales del contenido espectral de la aproximación improvisativa de Reichel. Además en esta versión de SEALI incluí 6 modelos individuales por cada uno de los instrumentos utilizados que categorizan las distintas técnicas instrumentales, supeditados a las predicciones estructurales que el modelo LSTM generaba, de manera tal que si la predicción estructural fuera más baja en términos de MFCCs y Flatness, se activan otros módulos de interacción, sumando más capas a la interactividad de SEALI.

Hablando de algunos aspectos de la partitura de esta obra, por momentos propone técnicas instrumentales muy acotadas, pero mediante trazos más libres, invita a los músicos a explorar la articulación y la duración de esos gestos, la dinámica y la energía musical desde la intuición y la escucha hacía todos los elementos sonoros que suceden. En otros momentos las indicaciones son mucho más libres, indicando solo la duración de los gestos, dejando que la capacidad creativa e inventiva de los músicos sea la que decida qué elementos utilizar y cómo y cuándo utilizarlos. A continuación comparto la partitura al pié de página.[8]

Para el proceso de ensamblaje sonoro-musical humano-máquina, partí desde una perspectiva muy amplia, de manera que los músicos fueran ellos quienes tomaran las decisiones del contenido musical en esta obra; desde sus propias memorias, metáforas, formas de aproximarse a su instrumento y su propio entusiasmo para improvisar. Asimismo, busqué incentivar su participación desde una escucha atenta individual y colectiva en interacción con SEALI. Fue interesante trabajar desde esta perspectiva ya que los invitaba a romper con sus esquemas sonoros musicales para intentar ir más allá de lo ellos mismos podían permitirse, dentro un contexto formal como el que se espera que ocurra en un concierto de esta naturaleza. Fue interesante que

[8] `https://archive.org/details/aaron-escobar-interdictos-protesicos`

también con la ayuda del director Rodrigo Cadet se haya abierto la posibilidad a que ellos pudieran explorar mucho más de lo que inicialmente estaban proponiendo, un proceso de escucha, adecuación que, tal vez, los llevó a un desprendimiento de su técnica más convencional (idiomática) y que marcó la pauta para que cada músico hiciera su propuesta gestual a través de la retroalimentación con todos los elementos sonoros que sucedían.

Dada la forma abierta de la pieza, el ánimo a la exploración y que no hay coincidencias explícitas que deban de hacer los músicos con SEALI, un elemento importante que le dio mayor maleabilidad a la obra fue que el director Rodrigo Cadet se apropió de la dirección temporal, pese a la línea de código temporal en segundos que se sugiere en la partitura. Esto me resultó fascinante ya que él mismo podía jugar con las velocidades de los gestos, subiendo o bajando la velocidad interpretativa mediante su dirección, respetando las coincidencias temporales que suceden entre los instrumentos.

En la plática retroalimentación que tuvimos en el primer ensayo fue valioso escuchar a Mariana y Carla llegar a la reflexión de que los materiales que ellas propusieron, no podían variar mucho debido a que el propio SEALI se estancaba en una sonoridad homogénea que a ellas mismas las llevaba a no intervenir en el proceso desde una exploración más arriesgada y heterogénea. A nivel técnico lo que estaba ocurriendo fue que para estos ensayos solo partí de un micrófono (en vez de 6) el cual se encontraba más cerca de las bocinas que el ensamble, generando así una retroalimentación más activa hacia lo que el propio SEALI propuso en términos sonoros; en lugar de tener una presencia más activa de los materiales sonoros que los músicos estaban proponiendo.

Debido a la premura del tiempo, ya que se tuvieron que montar 5 obras más en los ensayos, en el segundo ensayo, mientras que el ensamble y el director escuchábamos individualmente los materiales propuestos por cada músico, me permití hacerles comentarios a cada uno. De esto resultó un espacio de escucha activa compartida para trabajar los gestos sonoros propuestos y ahondar en las posibilidades que podrían explorar, más allá de las aproximaciones, en algunos casos, apegadas a una

exploración convencional de su instrumento. En el caso particular del percusionista, percibí cierto conflicto cuando le sugerí otras posibilidades en las que podría abordar el primer trino sobre el Gran Cassa que aparece en la partitura. A lo cual él me contestó que más bien yo le dijera puntualmente qué es lo que yo quería escuchar, o qué tocar específicamente, para así ahorrar tiempo de montaje y que yo estuviera conforme con el resultado. A lo cual le respondí que más allá de lo que yo quiera escuchar o no, realmente me interesa que ellos puedan proponer, la aproximación y los materiales, cuestionando las formas tradicionales de creación musical en la cual un compositor dicta a los músicos qué hacer, y más allá de eso, explorar otras formas de aproximarse a su instrumento y desde ahí ser parte de del proceso creativo, de manera que la propuesta de la obra siguiera abierta a esa libre exploración y creación al momento. Esto podría resultar un poco contradictorio para algunos, pero en las instrucciones de la partitura está planteado desde el inicio. Interdictos protésicos es una obra abierta pero tiene sus reglas y una de ellas es abrir la exploración hacia otras formas posibles de abordaje instrumental desde la propuesta sonora-musical de quienes la toquen.

El performance público ocurrió el 15 de octubre de 2022 ahí la magia de la impermanencia ocurrió, masas sonoras que van y vienen, puntos de coincidencia derivados de la emergencia, la escucha, el caos, el azar y la incidencia que cada uno aplicaba desde su propia agencialidad a la construcción de esta forma musical. Los materiales que se llegaron a producir incluso antes de que SEALI comenzará a interactuar con ellos, se asemejan mucho a las cualidades sonoras propuestas por SEALI. Una exploración muy al estilo de la improvisación libre comenzó a suscitarse, donde la aproximación a la frecuencia o el ritmo ya no era lo que más importaba, sino la cualidad sonora, el brillo, la centralidad, la complejidad y la densidad espectral fueron los elementos predominantes.

Debido a las características de esta pieza, el contraste con otras piezas del mismo concierto fue muy notorio. En principio la propuesta de integración de los propios músicos al proceso compositivo en tiempo real, detonó un espacio de libertad y complicidad desde los ensayos. Esto propició que cada uno desde su propia praxis musical rompiera con los esquemas en torno a su aproximación musical. Las exploraciones

que hicimos de manera individual durante los ensayos y la interacción con SEALI abonaron decisivamente a la forma y estética que esta obra tomó. Si bien el abordaje musical y creativo puede mejorar mucho, fue una primer aproximación que me deja con el ánimo para seguir explorando aproximaciones del estilo, que busquen cuestionar las formas tradicionales a las que los músicos están acostumbrados a explorar sus instrumentos y más allá de eso, delegar todo el peso y punto de partida a esa escucha profunda y a dejar abierta la invitación a ser partícipes en ese espacio de encuentro de creación. Interdictos Protésicos busca construir un proceso vivo que parte del cuestionamiento hacia las formas instauradas en las que nos aproximamos a la creación músical para dar lugar a ese espacio en construcción al cual los propios músicos sean los que den voz, colores, matices y temporalidades al momento de crear esta pieza colectiva. A continuación, comparto los resultados de este proceso: un ensayo previó, dos tomas del concierto en vivo y la grabación ambisónica del mismo:

5.3.2. Movil II con SEALI V.0.2

Para versión del Movil II de Manuel Enríquez, trabajé con dos modelos anidados, el primero hace predicciones sobre qué podría ocurrir dada una serie de elementos musicales y el segundo clasifica el contenido espectal –idéntico versión de SEALI V.0.1–, de manera que los elementos tímbricos detectados están supeditados a las predicciones a nivel estructural del modelo LSTM. El modelo que analiza la estructura, está entrenado con la interpretación de Fabián Rangel, de este modo dada una secuencia de 100 segmentos hace sus predicciones, que son la base para decidir si su reacción estará condicionada por una predicción de las secuencias o una clasificación espectral.

En las figuras 5.7 y 5.8 es posible observar los segmentos originales del 100 al 200 de Movil II y la predicción que el modelo hace al entregarle los 100 primeros fragmentos con los que ha entrenado.

Para medir el índice de certeza de este modelo dividí la base de datos en dos partes 66 por ciento para entrenamiento y el otro 33 por ciento para pruebas obteniendo una perdida de 0.000573, y una certeza de 1.0. En el siguiente cuaderno de Google Colab es posible hacer la comprobación de este experimento. Además en la figura 5.9

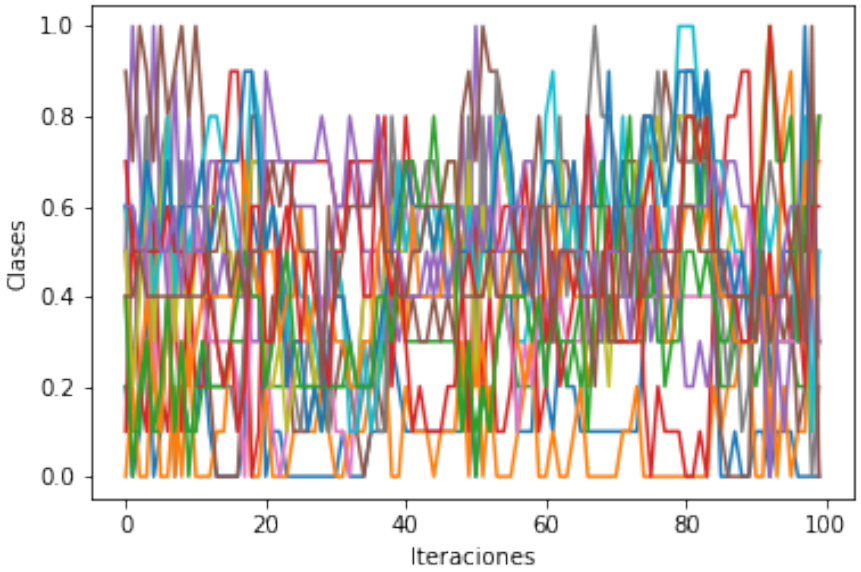

Figura 5.7: Segmentos originales del 100 al 200 de Movil II.

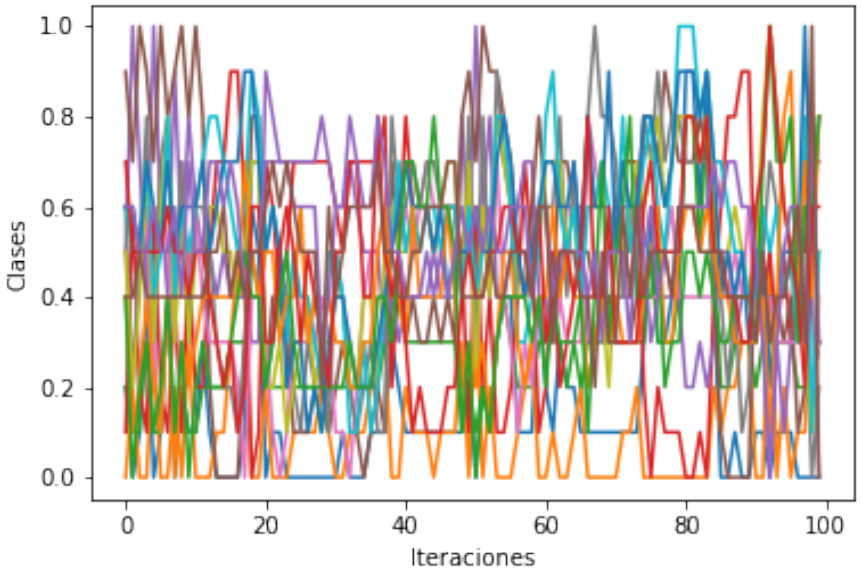

Figura 5.8: Predicción de los segmentos subsecuentes recibiendo como entrada los primeros 100.

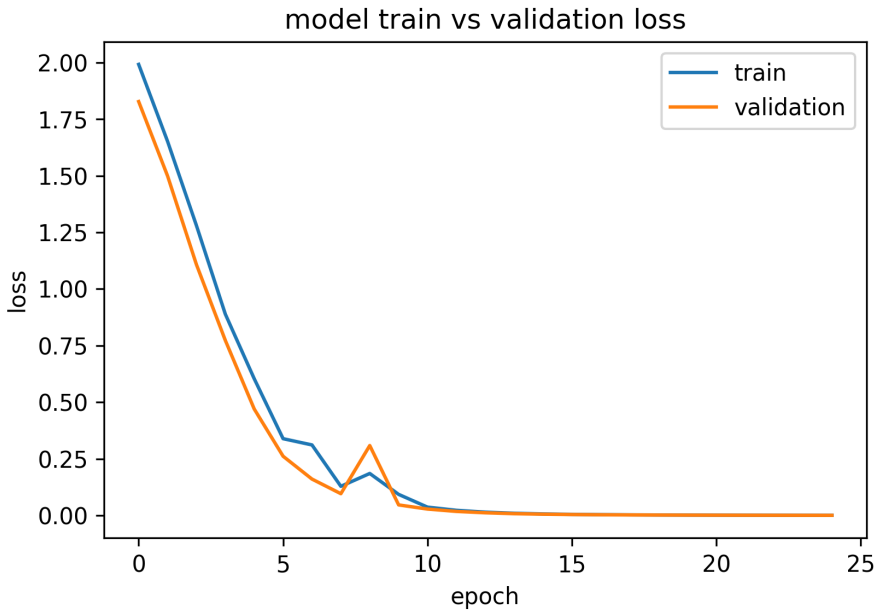

Figura 5.9: Predicción de los segmentos subsecuentes recibiendo como entrada los primeros 100.

es posible observar el comportamiento del modelo al entrenar y validar la información presentada.[9]

A continuación presento dos grabaciones de esta obra, la primera fue en ensayo y la segunda fue la versión que se presentó el día 18 de octubre de 2022 en el auditorio del MUAC dentro del marco del Foro Internacional de Música Nueva Manuel Enríquez.

`https://archive.org/details/movil-ii-muac-seali`

[9] `https://bit.ly/3UDQmTi`

Capítulo 6

Conclusiones

En esta investigación he expuesto diversas metodologías que permiten explorar tecnologías como el aprendizaje y la escucha de máquina aplicadas al análisis de bases de datos de improvisaciones libres. Estos desarrollos me permitieron crear un sistema sonoro que escucha e interactúa en tiempo real consigo mismo o con otros músicos, adaptando sus respuestas al contexto sonoro de acuerdo con la base de datos de entrenamiento. La configuración del sistema contempla dos fases, la primera se concentra en el análisis espectral del sonido e incluye varios procesos algorítmicos: la captura de una fuente sonora que es transformada en audio digital, la segmentación del audio, el análisis espectral a través de la extracción de características, la clasificación sin supervisión para crear una base de datos y la creación con redes neuronales profundas de un modelo supervisado aplicado a la predicción sonora en tiempo real. La segunda se centra en el análisis estructural musical a varios niveles y contempla: el análisis de descripciones numéricas del audio en forma de series de tiempo mediante el algoritmo LSTM para aproximarse a la identificación de estructuras musicales dentro de la práctica particular de algunos improvisadores libres. A su vez, las interacciones de este sistema conjuntan tres funciones. El primero se concentra en la predicción, búsqueda y reproducción, de archivos de audio midiendo el índice de similitud comparando entre lo escuchado y la base de datos de entrenamiento. El segundo, también realiza la predicción y búsqueda de archivos similares que envía a través del protocolo OSC para ser utilizados por Supercollider y aplica distintos procesamientos a los audios se-

leccionados por el algoritmo de predicción. Por último, el tercero envía los datos de las predicciones a Supercollider y utiliza esa información para modificar en tiempo real los parámetros de varios sintetizadores o *ugens*. En este sentido, el tipo de interacción de SEALI se encuentra en dos esquemas: la sonificación, donde los datos de clasificaciones y predicciones son aplicados para cumplir con diferentes funciones booleanas y mapeados para modificar los estados internos de diferentes sintetizadores, y el esquema de reproducción de archivos de audio de diferentes bases de datos. Este último esquema responde de manera iterativa al siguiente planteamiento: dada una serie de datos de entrada, qué es lo que sigue a continuación. Estos algoritmos fueron implementados bajo una arquitectura cliente/servidor mediante Python, Supercollider y TensorFlow Serving, usando librerías como tensorflow, keras, scikit-learn, essencia, scmir, pydub, pyaudio, pythonosc, toolz, request, entre otras. En consecuencia, uno de los objetivos del presente trabajo es ampliar la definición de las prácticas artísticas mediadas por computadoras desde una perspectiva que incluya, además de los aspectos de producción tecnológica y representación simbólica del audio, a la escucha. Se trata de una aproximación a la música generativa, aplicando técnicas de aprendizaje automático para llevar a cabo un proceso de escucha activa de un sistema generativo para obtener modelos descriptivos que den cuenta de esa escucha artificial. Los modelos descriptivos generados que parten del análisis del contenido espectral y las formas improvisativas, se despliegan a modo de clasificador y predictor de improvisaciones libres para generar nuevas comprovisaciones generativas y, por lo tanto, un modo muy particular de escucha automática.

Cabe destacar que SEALI en distintas facetas de desarrollo, fue probado con improvisadores solistas y varios grupos de músicos que retroalimentaron el proyecto y aportaron materiales sonoros que sirvieron para entrenar modelos computacionales adaptados específicamente a su aproximación musical, su práctica dentro de la improvisación libre y la comprovisación. Además ellos mismos hicieron posible la realización del ciclo de conciertos *resistencias maquínicas* que dio cuenta de los procesos de colaboración humano-máquina en contextos de conciertos de improvisación libre. El resultado fue muy positivo ya que permitió generar nuevos materiales musicales y adaptar cada vez más los recursos y posibilidades de SEALI a contextos reales

de interacción humano-máquina. Desde esta perspectiva SEALI tuvo cierta relación dentro del ámbito de la música experimental o los espacios de improvisación libre en la Ciudad de México que me permitieron retroalimentar de la escena local para llevar el desarrollo de SEALI hacia lugares más interesantes que le llevaran dialogar de formas más coherentes y novedosas con los músicos improvisadores. También hubo colaboraciones y apariciones en espacios como Toplap Barcelona y Hangar en la misma ciudad que me ayudaron a posicionar el proyecto fuera de México y tener una retroalimentación sumamente valiosa por parte de comunidades interesadas en el desarrollo tecnológico, la creación musical y particularmente el livecoding. En su fase final, tuvo presencia activa en dos conciertos del Foro Internacional de Música Nueva Manuel Enríquez, así como en algunos foros independientes en la ciudad de Aguascalientes donde actualmente radico. Si bien los desarrollos actuales de SEALI los he generado de manera individual a través de toda la retroalimentación que la práctica improvisativa y la investigación me han proporcionado, cabe mencionar la importante participación en etapas tempranas dentro su desarrollo de mi querido amigo, compositor y programador Diego Villaseñor y el comité tutor integrado por Hugo Solis, Ivan Paz y Caleb Rascón.

El sistema está disponible para su instalación y uso bajo licencias libres en el siguiente repositorio github: `https://github.com/Atsintli/SEALI-V.I`. Este repositorio está bajo una licencia GNU General Public License V. 3 y estará permanentemente en transformación conforme SEALI vaya adquiriendo nuevas funcionalidades. La idea de que permanezca de este modo es para invitar particularmente a futuros alumnos de la maestría y doctorado en tecnología musical, improvisadores libres y público en general a colaborar de manera activa de este repositorio con el objetivo de crear una comunidad activa en torno a este proyecto, sus posibles derivados futuros y particularmente en torno a las técnicas de escucha y aprendizaje de máquinas enfocados en la creación musical. Asimismo las bases de datos utilizadas quedan abiertas y disponibles para pruebas o experimentos futuros que se deseen realizar. Por otro lado, cabe destacar que esta tesis intenta colocar sobre la mesa dentro de la institución en la que se inserta este trabajo, un referente que busca construir y sumarse a las líneas de trabajo en torno a la creatividad musical y la inteligencia artificial

que actualmente se desarrollan en el posgrado de tecnología musical de la facultad de música en la UNAM.

La principal aportación introducida con SEALI, es que, este marco de trabajo permite, por el simple mecanismo de dividir cada tarea en subrutinas, integrarlas en la resolución de múltiples tareas que pueden ser de utilidad tanto para músicos y no músicos, improvisadores, investigadores musicales y programadores. Además de contar con la posibilidad, dependiendo del proyecto en cuestión y según sea necesario, de derivar cada una de las subrutinas en distintas aplicaciones. Otra cuestión es que para improvisar con SEALI se requiere de un nivel relativamente bajo de recursos computacionales para ejecutar los modelos creados. Sin embargo, construir esos modelos no fue tarea fácil, ya que para esto fue necesario trabajar con algunos servidores del IIMAS en la UNAM, facilitados por el Dr. Caleb Rascón, los cuales tampoco me permitieron hacer experimentos de mayor alcance debido a la falta de memoria para trabajar con bases de datos y arquitecturas neuronales más grandes. Así, nos posicionamos ante un nuevo tipo de tecnología, que si bien es accesible para cualquier persona que cuente con una computadora e internet, solo las grandes empresas (en algunos casos universidades), pueden tener acceso a hardware especializado para realizar modelos mucho más grandes y generales que puedan trabajar con billones de datos y neuronas para entrenar sus modelos. Cabe mencionar que a lo largo de esta investigación estuve tratando por diversos medios de acceder a alguno de los clusters de supercómputo que poseen algunas instituciones en México, pero esto no fue posible debido a la falta de información y ausencia de convocatorias que incentiven acceso a los clusters, no solo a proyectos científicos sino también a proyectos vinculados al arte, la ciencia y la tecnología. Definitivamente contar con un poder de cómputo lo suficientemente grande permitiría extender las capacidades de aprendizaje y memoria de SEALI.

Con todo ello, esta investigación puede ser vista como una plataforma que intenta problematizar la relación entre arte, ciencia y tecnología. Mediante su diseño, sugiere oportunidades para el uso de herramientas tecnológicas aplicadas a la construcción de nuevas experiencias artísticas y otros usos posibles de la tecnología, particularmente el aprendizaje y la escucha de máquina. Para abordar el problema de la creciente

penetración tecnológica en el campo del arte, he buscado emplear las herramientas tecnológicas no para reproducir o copiar lo que otros desarrolladores o músicos han hecho, sino emplear estas herramientas y conocimiento para construir nuevos imaginarios. Aunque sigue habiendo algunos aspectos por explorar de SEALI, como trabajar con otras bases de datos, arquitecturas de redes neuronales más grandes, ampliar los grados de memoria en los que puede llevar un seguimiento sobre los momentos que han transcurrido en una improvisación, aplicar otras técnicas del aprendizaje automático como la atención (transformers) o hacer un desarrollo más profundo sobre los mecanismos que emplea para interactuar con otros improvisadores, quedo conforme con los resultados generados hasta el momento. Si bien, SEALI aún no es capaz de detectar elementos sonoros referentes a un contexto específico u otras prácticas musicales diferentes a la improvisación libre, su grado de agnosticismo para reaccionar a todo aquello que escucha me parece un aspecto interesante que le da la posibilidad de activar sus mecanismos sonoros sin tener una limitante o un sesgo particular más allá de las constricciones booleanas empleadas en los procesos de sonificación de datos y reproducción de archivos de audio.

Una de las discusiones que aún quedan abiertas en torno al desarrollo de SEALI es la falta de accesibilidad para ser utilizado por otras personas ajenas al mundo de la programación. Si bien, tiene la posibilidad de réplica ya que los desarrollos están hechos bajo licencias libres y *copyleft* y código abierto, aún hace falta implementar una simplificación al motor del sistema, es decir, no utilizar múltiples módulos y programas para producir un resultado sino utilizar software para para gestionar los módulos en un servidor con herramientas como Docker o Kubernetes, con el objetivo de simplificar su instalación y estar listo para usarse. En este sentido y al explorar varias opciones, encuentro que una alternativa sería utilizar el lenguaje de programación JavaScript el cual tiene la posibilidad junto con el protocolo html y .css de montar todo el sistema en una página web. Naturalmente esta aproximación conlleva sus propias limitaciones como la calidad o fluidez sonora pero posibilita la apertura y usabilidad del sistema a personas que cuenten con acceso a Internet y una computadora. Es importante mencionar que los alcances de esta investigación y los desarrollos tecnológicos planteados están limitados a los intereses y aplicaciones aquí expues-

tas. Desde el principio me posiciono como investigador, músico e improvisador con el interés de indagar ciertas áreas del aprendizaje automático y la escucha artificial y aplicarlas para entender y expandir las posibilidades creativas de la creación musical, específicamente dentro de la práctica de la improvisación libre. Así me posiciono como un músico que explora estos territorios partiendo de la escucha atenta para crear herramientas que dentro de estas prácticas me son de interés y utilidad. En ningún momento pretendí plantear el desarrollo de nuevas funciones algorítmicas ni de plantear una contribución directamente en ese campo, lo que busco es propiamente hacer uso de los recursos tecnológicos existentes en esta área para intentar generar perspectivas estéticas muy particulares en torno a la creación musical y particularmente el análisis tímbrico y estructural de/en la improvisación libre.

Un trabajo, o experimentos a futuro que me quedan por explorar es la creación de modelos computacionales que respondan a otros géneros musicales tal vez desde la tonalidad o desde una perspectiva rítmica más rigurosa, si asumimos que está la improvisación libre es más compleja, estos otros modelos podrían adaptar sus funciones e interactuar con músicas dentro de otros géneros, como el blues, el jazz o el free jazz. Otra idea interesante a explorar es que la interacción de SEALI pueda ser extendida hacia agentes más que humanos, por ejemplo, al inicio de esta investigación utilicé una máquina sonora para crear interacciones con aves en el bosque, respondiendo mediante escucha de máquina a las alturas que ellas producían, a lo cual tanto la máquina como las aves parecían estarse comunicando a través de los reflejos sonoros. La idea de integrar el entorno me resulta muy atractiva debido a que este puede transformar la forma en la que la máquina interactúa y poder ir más allá de las estéticas de una práctica o estilo particular.

Paralelo a esto, en el capítulo uno, documenté algunos proyectos que parten de tecnologías como el aprendizaje y la escucha de máquina para crear diversos sistemas interactivos musicales, destacando los enfoques metodológicos empleados. Esta revisión fue de utilidad para conocer el estado actual de algunas aproximaciones derivadas de la experimentación creativa en contextos de improvisación libre, y la relación que éstas guardan con el desarrollo de herramientas tecnológicas aplicadas a la creación musical en tiempo real.

Por otro lado, a través de las reflexiones plasmadas en esta tesis encuentro que el punto nodal que conecta cada una de ellas es el acto de la creación en sus múltiples manifestaciones, donde la ruptura y los cuestionamientos hacia la técnica de las estructuras hegemónicas son el punto de partida que detona gran parte del imaginario creativo aquí plasmado.

> Montero y Donoso examinan determinadas tácticas de resistencia implementadas en la región [de latinoamérica] desde la convergencia entre el arte, la ciencia y la tecnología para dar cuenta de modos "bastardos" de funcionamiento en relación con los contextos hegemónicos: proyectos que (d)enuncian o señalan; proyectos que deconstruyen y desarman; y proyectos que proponen e inventan alternativas. (Adler *et al.*, 2021)

Asimismo, en la tesis se presenta al free jazz como un movimiento de resistencia social que busca oportunidades más justas para vivir y expresarse al crear una ruptura con los modelos estéticos hegemónicos de occidente.[1] Desde este espíritu de ruptura, intento colocar una serie de cuestionamientos dinámicos y estéticos que promuevan el surgimiento de otras formas de creación musical; la improvisación libre o la comprovisación y su entrelazamiento con otros usos posibles del aprendizaje y la escucha de máquinas. Por otro lado, el acto creativo en la programación que parte de una perspectiva híbrida, abierta al cambio, la adaptación y la apropiación de herramientas intenta dar lugar a modos de aprendizaje intuitivo con los recursos disponibles en las distintas plataformas de la red.[2] Desde esta relectura y aplicaciones otras de las tecnologías, giran nuevas tendencias tecno–artísticas en torno al aprendizaje de máquinas que no solo buscan aplicar estas herramientas a sus fines artísticos sino que también hay un intento por entender de manera profunda su funcionamiento.

[1] El "jazz europeo" en sí comenzó como una internalización europea de la hegemonía cultural estadounidense que combinó todas las historias, idiomas y estilos combinados del continente en un solo monolito. (Lewis, 2008)

[2] Si bien muchas de las plataformas que permiten acceder al conocimiento o utilización de estas herramientas son privativas y provienen de ese pensamiento capitalista hegemónico por excelencia, somos los mismos desarrolladores, programadores, analistas, críticos y creativos los que estamos generando contenidos que permiten indagar a cabalidad sobre cómo estas herramientas funcionan, desde ese acto de abrir el código para que otras personas podamos acceder a estas herramientas y conocimiento colectivo. El otro lado de la moneda que resulta interesante analizar es el fuerte interés de estas empresas por seguir abriendo su código, en algunos aspectos, e incluso generando economías en torno a los desarrolladores de estas iniciativas para acelerar el desarrollo de herramientas útiles a sus propósitos.

Así estos “modos "bastardos"de funcionamiento respecto a los espacios hegemónicos” (Adler *et al.*, 2021), permiten desarticular los automatismos heredados de la relación con las máquinas, a través del empoderamiento híbrido hacia la tecnología. Ello permite la realización de proyectos artísticos que cuestionan las dinámicas de un capitalismo salvaje y coadyuvan a tejer otro tipo de relaciones humanas –con o sin prótesis tecnológicas– e híbridas.

> “Por un lado, accedemos a la posibilidad de comprender el secreto de las máquinas, de entender el programa que opera en la llamada “caja negra”, [...]. Esta posibilidad de superar la lógica fetichista inherente a la producción tecnológica que el mercado ha naturalizado se muestra en abierto contraste con la advertencia de Žižek sobre la tecnología como algo cada vez más gris e incomprensible [13].[...] Al respecto, la apropiación de esa tecnología, desobedeciendo sus determinaciones de fábrica, permite la producción de nuevas significaciones situadas, personales, arbitrarias o poéticas. Y desde ahí es posible subvertir las determinaciones económicas implícitas en el diseño de los artefactos tecnológicos, como por ejemplo su veloz obsolescencia.” (Adler *et al.*, 2021)

Estas ideas detonan el interés por desarrollar el presente proyecto, que recae en empoderarnos como artistas de las capacidades tecnológicas actuales relacionadas con el aprendizaje automático y la escucha de máquinas para entenderlos a cabalidad, conocer sus alcances y acercarnos de manera informada a su uso e implementación. Esta perspectiva puede ayudar a transformar tanto la propia práctica de producción artística, así como aportar nuevas perspectivas para aproximarnos de manera crítica hacia el sistema maquínico imperante que está condicionado a efectuar mecanismos productivos, capaces de repetir interminablemente y de manera disciplinada cualquier tarea. Así, el humano no es un agente aislado sino que opera en un ambiente de colectividad compuesto por humanos y más que humanos, del cual las máquinas evidentemente son parte. El humano y la computadora se encuentran entrelazados en un continuo flujo de intra-acciones e intercambio de conocimiento a múltiples niveles, que finalmente lo llevan a un ciclo de afectaciones agenciales producto de la mediación técnica. Esta tesis es un intento por repensar esas afectaciones agenciales y cuestionar la idea

de que las capacidades del aprendizaje de máquina solo siguen alimentando nuestra propia alienación social, siendo los elementos reificados dentro de ese mecanismo, hasta el punto en el que las máquinas adquieren cierta autonomía para modificar y reedificar las interacciones humanas en sus múltiples aspectos; tanto en la individuación como en la colectividad sin que que formemos parte activa de ese cambio. Desde esta problemática, más bien busco proponer formas de aproximación activa hacia la utilización de estas tecnologías, a través de su estudio–intervención–modificación–apropiación y pensarlas como herramientas capaces de romper con nuestros propios hábitos para formar otros espacios de acción. Al desdibujar cada vez más las fronteras entre esa tecnología "gris e incomprensible" (Adler *et al.*, 2021) y "opaca, [...] totalmente incomprehensible, incluso inaccesible" (Žižek, 2010), para posibilitar el surgimiento –desde la mutua afectación– de nociones más comprometidas ligadas a los procesos de su utilización para dar lugar a nuevas formas escultóricas que nuestros cuerpos-mentes-dispositivos toman en ese proceso.

De ahí es posible seguir derivando múltiples aproximaciones que promuevan la creación sonora y musical desde otras perspectivas, donde la toma de decisiones ya no puede ser pensada desde una individualidad inherente propiamente a lo humano, sino a partir de las múltiples capas y tipos de inteligencia que comenzamos a ver con el advenimiento del análisis masivo de datos que las supercomputadoras actuales pueden realizar.[3] Cada decisión a la que llega un modelo entrenado bajo este esquema, se encuentra influida por una cantidad enorme de sesgos provenientes de agentes naturales y artificiales (presentes y ausentes) afectando el sistema en cuestión. Así, la capacidad de la toma de decisiones de las máquinas no excluye el carácter humano, las máquinas, en todo caso, son dispositivos que ayudan a conjuntar un poder de conocimiento –cada vez más despersonalizado– que contiene gran parte del conocimiento de la humanidad. Las máquinas no son dispositivos a los cuales podamos atribuirles caracteres o propiedades humanas o sociales, aunque intenten hacernos creer lo contrario. Un ejemplo reciente sobre este tema, es la controversial entrevis-

[3]Ejemplo de ello podrían ser sistemas como el GPT-3 de google el cual ha demostrado resolver tareas bastante complejas como escribir código computacional o tener una asombrosa conversación, sin estar explícitamente programado para tales fines. En sus propias palabras: "Puedo entender nuevos conceptos y problemas al relacionarlos con cosas que ya he aprendido." `https://www.youtube.com/watch?v=PqbB07n_uQ4&ab_channel=EricElliott` Otro sistema similar sería GPT-NEO[4], este se encuentra disponible bajo licencias libres para su uso.

ta que entabló un científico de google con la inteligencia artificial *LaMDA*, la cuál asegura a través de sus respuestas, tener una conciencia y sentimientos similares a los humanos.[5] No podemos afirmar que un sistema que basa sus respuestas en activaciones de redes neuronales masivas; operaciones numéricas, pesos y sesgos, sea un criterio para determinar si una máquina tiene o no consciencia y ha traspasado la prueba de Turing, cuando en realidad más bien es capaz de dar respuestas que pretenden ser inteligentes. ¿Cuál es el límite de la prueba de Turing y hasta dónde nos puede llevar la inventiva de sistemas y pruebas mucho más sofisticadas? Ante la promesa de productividad, eficiencia, autonomía y ahora creatividad inédita, en función de las resoluciones que pueden tener máquinas como GPT-3 (u otros megatrones especializados en el modelado de lenguaje natural autorregresivo) o *Stable Diffusion* (especializado en convertir texto a imágenes) para efectuar la realización de múltiples tareas, es importante plantearse cómo imaginamos posibles escenarios en torno a las máquinas que paulatinamente configuran de una forma muy particular las relaciones humano–humano–máquina. En algunos casos, generando un vínculo tan estrecho con los dispositivos que es imposible concebir su ausencia; ello no deja de lado que estos dispositivos nos sirvan como herramientas capaces de redirigir esa atención depositada en ellos hacia espacios de introspección, autocuestionamiento y reflexión, para encontrar un balance entre lo que las máquinas pueden hacer por/en nosotros y el tipo de relaciones agenciales que producen en cada ámbito donde se insertan. En un futuro no muy lejano, no será extraño poder acceder a sesiones de psicoanálisis virtuales con avatares donde una máquina que escuche y pueda interactuar con una atención y un cuidado tan delicados como nuestra personalidad sea en un momento particular. La asertividad que estas máquinas pueden generar no depende de las bases de datos empleadas para su entrenamiento, sino de las múltiples relaciones que pueden generar producto del nivel relacional con el que pueden entretejer una secuencia de entrada para generar salidas de forma novedosa. [6] Más allá de estas herramientas, es importate dejar sobre la mesa la pregunta de hacía dón-

[5]En el siguiente enlace es posible leer la entrevista completa. `https://s3.documentcloud.org/documents/22058315/is-lamda-sentient-an-interview.pdf`

[6]Un ejemplo muy interesante de características relacionales mediante el lenguaje, puede ser encontrado en las respuestas que GPT-3 hace a nueve filósofos quienes plantean diversos problemas y preguntas al modelo para analizar sus respuestas. `https://dailynous.com/2020/07/30/philosophers-gpt-3/`

de deberían ir encaminados los esfuerzos de seguir indagando y desarrollando estas tecnologías. Algunos ejemplos de ello serían, combatir o inducir activamente sobre el cambio climático o los desastres naturales o partir del propósito de revolucionar la creación de políticas públicas y económicas para combatir la desigualdad social y una serie de consideraciones bioéticas para garantizar su usabilidad en cuestiones de salud pública considerando sus ventajas y limitaciones así como y sus implicaciones regulatorias. Esto sería una visión muy positiva de hacia dónde podría dirigirse, pero la inteligencia artificial también puede dañar la democracia, el tejido social y las libertades civiles.

Así, esta investigación intenta desde sus múltiples facetas, abrazar la afectación como una propiedad intrínseca agencial necesaria para la activación de espacios sujetos a la intra-acción que tanto agentes digitales como humanos comparten en una permanente incidencia. Con esto en mente resulta preciso, encontrar un estado de relación más justo con los dispositivos tecnológicos y seguir con los cuestionamientos, a través de esta y otras propuestas de apropiación tecnológica, de aquellos elementos externos que determinan nuestras posibilidades de relación hacia una sola forma de experimentar y concebir la tecnología, generalmente desde una lógica heredera del "racionalismo modernista europeo, que insistía en la legitimidad simbólica de un modelo de sociedad capitalista."(Adler *et al.*, 2021) Es por ello que no apelo al uso de herramientas privativas para crear los desarrollos tecnológicos que se desprenden de esta tesis, sino a la posibilidad de seguir aportando otras perspectivas ligadas al uso del software libre y el código abierto, así como abrir la posibilidad de réplica a otros artistas o programadores sobre los desarrollos tecnológicos de este trabajo.

Desde la perspectiva musical apuesto por las búsquedas ligadas a desmantelar el estado más predominante que existe en las formas de creación musical dentro de la academia[7], para atreverme a cuestionar las prácticas y herramientas que hemos heredado y mirar hacia la multiplicidad que el código, el pensamiento algorítmico y la creación sonora partiendo desde la escucha profunda en interacción con la in-

[7]Al respecto Henry Cawell menciona: "Las reglas de armonía, contrapunto y orquestación actualmente enseñadas ciertamente no sugieren al estudiante materiales adaptados a sus propios deseos expresivos, sino que se le da un pequeño y circunscrito conjunto de materiales, ya muy utilizados, junto con un conjunto de prohibiciones para aplicarlas, y luego se le pide que se exprese sólo dentro de estas limitaciones".(Lewis, 2008)

vestigación y la improvisación libre pueden proponer. Así desde esta perspectiva, el desarrollo tecnológico, la investigación, y la práctica artística de este trabajo entran en un continuo ciclo de retroalimentación, en ese devenir de intra-acciones donde cada una de ellas se ve afectada y construida al mismo tiempo por las demás. Desde ahí es posible pensar en que más allá de los fines y aplicaciones originarias que cada una de estas áreas tienen por separado, la afectación que cada una de ellas produce sobre las demás abre un espectro de posibilidades inédito. Particularmente desde el arte es posible pensar en que los desarrollos tecnológicos no necesariamente deben tener un funcionamiento óptimo o una validación rigurosa en términos científicos para producir resultados estéticos o planteamientos interesantes, sino que es posible emplear esta falta de rigor para intentar transformar la misma práctica artística así como las utilidades finales de estas herramientas, y, de este modo, ampliar el panorama de acción de cada área por separado y en conjunto.

Bibliografía

Adler, J., Donoso, P., Farneda, P., Gontijo, J., Yeregui, M., y Montero, V. (2021). *Desmantelando la máquina: Transgresiones desde el arte y la tecnología en Latinoamérica*. Arte Contemporáneo / Arte Digital. Neural.

Allauzen, C., Crochemore, M., y Raffinot, M. (1999). Oracle: A new structure for pattern matching. En *International Conference on Current Trends in Theory and Practice of Computer Science*, volumen 1725, pp. 295–310.

Alpaydin, E. (2004). *Introduction to machine learning*. Adaptive computation and machine learning. MIT Press.

Bailey, D. (1980). *Improvisation. Its Nature and Practice in Music*. Moorland.

Barad, K. (2007). *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press.

Barad, K. (2011). Nature's queer performativity. *Qui Parle*, 19(2):121–158.

Bohm, D. y Apfelbaume, J. (1998). *La totalidad y el orden implicado*. Colección nueva ciencia. Editorial Kairós SA.

Braunstein, N. A. (2011). *El inconsciente, la técnica y el discurso capitalista*. Siglo XXI.

Brownlee, J. (2017). *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery.

Canonne, C. (2019). Listening to improvisation. *Empirical Musicology Review*, 13:2.

Collins, N. (2011). SCMIR: A SuperCollider music information retrieval library. En *Proceedings of the International Computer Music Conference 2011*, pp. 499–502.

Collins, N. (2016). Towards machine musicians who have listened to more music than us: audio database-led algorithmic criticism for automatic composition and live concert systems. *Computers in entertainment.*, 14(3):2.

Comaniciu, D. y Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Deutsch, D. y Sempau, D. (1999). *La estructura de la realidad*. Anagrama / Argumentos. Anagrama.

Dhomont, F. (1984). Michel Chion. Guide des objets sonores. Préface de Pierre Schaeffer. *Canadian University Music Review / Revue de musique des universités canadiennes.*, 1(5):366–368.

Dudas, R. (2010). Comprovisation: The various facets of composed improvisation within interactive performance systems. *Leonardo Music Journal*, 20:29–31.

Escobar Castañeda, A. A. (2016). Máquinas sonoras: aplicaciones de las ciencias de la complejidad a la creación musical y sonora. Tesis de licenciatura, Universidad Nacional Autónoma de México.

Escobar Castañeda, A. A. (2018). Hacia una escucha automaética de la espontaneidad: relaciones complejas entre la libre improvisacioén, la escucha y los sistemas de aprendizaje automaético. Tesis de máster, Universidad Nacional Autónoma de México.

Fiebrink, R., Trueman, D., Britt, C., Nagai, M., Kaczmarek, K., Early, M., Daniel, M., Hege, A., y Cook, P. (2012). Toward understanding human-computer interaction in composing the instrument. *ICMC*.

Fiebrink, R. A. (2011). *Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. Tesis doctoral, Princeton University, USA.

Galiana Gallach, J. L. (2018). De la naturaleza de la improvisación libre: elementos esenciales para su identificación y diferencias con la composición escrita. *Itamar. Revista de investigación musical: territorios para el arte*, 0.

Gómez, E. (2006). *Tonal Description of Music Audio Signals*. Tesis doctoral, Universidad Pompeu Fabra.

Hochreiter, S. y Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Johnston, J. D. (1988). Transform coding of audio signals using perceptual noise criteria. *IEEE on Selected Areas in Communications*, 6:314–323.

Latour, B. y Aúz, T. (2001). *La esperanza de Pandora: ensayos sobre la realidad de los estudios de la ciencia*. CLA·DE·MA.: Sociología. Gedisa.

Levy, B. (2004-2012). *OMax The Software Improviser*.

Lewis, G. (2008). *A Power Stronger Than Itself: The AACM and American Experimental Music*. The Kenneth Nebenzahl, Jr. , Lectures in the History of Cartography Series. University of Chicago Press.

Lewis, G. E. (1996). Improvised music after 1950: Afrological and eurological perspectives. *Black Music Research Journal*, 22:215.

Matthews, W. (2012). *Improvisando: La libre creación musical*. Turner Publicaciones.

Morin, E. y Pakman, M. (1998). *Introducción al pensamiento complejo*. Ciencias cognitivas. Gedisa.

Najmanovich, D. (2008). *Mirar Con Nuevos Ojos Nuevos Paradigmas en la Ciencia Y Pensamiento Complejo*. Colección Sin fronteras. Biblos.

Nancy, J. (2008). *A la escucha*. Amorrortu Editores España SL.

Oliveros, P. (2005). *Deep Listening: A Composer's Sound Practice*. iUniverse.

Painter, T. y Spanias, A. (1997). A review of algorithms for perceptual audio coding of digital audio signals. Available from `www.eas.asu.edu/ speech/ndtc/dsp97.ps`.

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, Icram.

Rancière, J. y Vila, J. (2012). *El tiempo de la igualdad: Diálogos sobre política y estética*. Pensamiento Herder. Herder Editorial.

Rebala, G., Ravi, A., y Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer International Publishing.

Rutz, H. (2016). Agency and algorithms. *Journal of Science and Technology of the Arts*, 8:73.

Schaeffer, P. y de Diego, A. (1996a). *Tratado de los objetos musicales*. Alianza música. Alianza.

Schaeffer, P. y de Diego, A. (1996b). *Tratado de los objetos musicales*. Alianza música. Alianza.

Schankler, I., Smith, J. B. L., François, A. R. J., y Chew, E. (2011). *Emergent Formal Structures of Oracle-Driven Musical Improvisations*, pp. 241–254. Springer Berlin Heidelberg.

Serra, M.-H. (1993). Stochastic composition and stochastic timbre: Gendy3 by iannis xenakis. *Perspectives of New Music*, 31(1):236–257.

Solís, H. (2006). Understanding Collective Gestural Improvisations; a Computational Approach. Tesis de máster, Universitat Pompeu Fabra.

Sutskever, I., Vinyals, O., y Le Quoc, V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Van Nort, D., Oliveros, P., y Braasch, J. (2013). Electro/acoustic improvisation and deeply listening machines. *Journal of New Music Research*, 42(4):303–324.

Vinet, H., Herrera, P., y Pachet, F. (2002). The cuidado project. En *3rd International Society for Music Information Retrieval (ISMIR) Conference*, pp. 197–203, Paris, France. Content Based Retrieval.

Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.

Xenakis, I. (1992). *Formalized Music: Thought and Mathematics in Composition*. Harmonologia series. Pendragon Press.

Žižek, S. (2010). Cyberspace, or the virtuality of the real. *jcfar*.

Švankmajer, J., Castro, E., Guiard, S., y Roman, D. (2014). *Para ver, cierra los ojos*. Pepitas de calabaza.