absorb the present status of the system over the last 60 seconds at a glance.

Property	-60s	-50s	-40s	-30s	-20s	-10s	NOW
currentDelayMs	10	10	11	10	10	11	9
jitterBufferMs	4	5	5	4	4	5	4
jitterReceived	19	20	23	18	17	27	20
RTT	19	19	19	19	19	18	19
packetsLost	29	29	29	29	29	29	29
packetsSent	6330	6380	6429	6480	6530	6581	6682
bytesSent	162138	163388	164613	165888	167138	168434	169754
bytesReceived	232019	233381	234743	236105	237267	238829	240191
requestsSent	202	203	204	205	206	207	208
audioInputLevel	12	5	12	9	30	91	29
audioOutputLevel	80	67	57	67	9	23	25

3;KSM6(AB! M=)*6,=; .2! ,+ .2>*= (+! 64* : 4*!)* . *=)*=;)! 67! . ! 3+*> 8 6,9+-! (4)=; ! : +,76, (.3 ; +B!

K</</ LA:?"*;C$)B7%*$;")4%)*4D79"*

&4)=; ! =376, (.*=63! ,+*,=+C .2! : ,= (. ,=2<! * .9+)! 63+! 67! 764,! 76, ( )! TUYVE! = (. ?+)/!)4 ; 0! .)! . !) ; 6, +K!)< (562=; /!)4 ; 0! .)! &%N%K! (+* . - . * ./!)4 ; 0! .)!*0+!=3)*,4 (+3* . *=63!6,!936 8 2+ - ?+! .564*!*0+! : +,76, (.3 ; +! .3 - !=*)!+3C=,63 (+3*K! .3 - ! - = ?=* .2! .4> - =6B! %3! 3+* 8 6,9+ - ! (4)=; /! 8 +! (=?0*! ; 2 .))=7<! *+2+ (+*,<! .)! (+* . - . * .B! &4)=; !=376, (.*=63! ,+*,=+C .2! : +, (=*)!4)!*6! (.9+! - + ; =)=63)!) : + ; =7= ; .22<! : +,* . =3=3?! *6! (4)=; ! .3 - ! : +,76,> (.3 ; +E!=B+B/! . !) ; 6, +! : ,6C= - +)!=3)*,4 ; *=63)!63!06 8 ! .3 - ! 8 0+3! *6! + (=*!)643 -B! &%N%! .3 - !)= (=2 . ,!36*+>+C+3*!)<)*+ ( )!*+22)! 4)! 8 0 . *! +2+ ; *,63=; !)=?3 .2)! 0 . C+! 5++3! + (=**+ -/! .3 - ! =376,> (.*=63! .564*! - = ?=* .2! .4 - =6! ?=C+)!4)! : +,76, (.3 ; +! ; 4+)! .3 - ! (. ,9+,)B! "4 ; 0!=376, (.*=63!=)!0=?02<!4)+742!=3! - +*+, (=3=3?! ;63*+A*/! : . ,*=; 42 . ,2<! 8 0+3! (4)=; = .3)! . ,+! ,+ (6*+!7,6 (!63+! .36*0+,B!

TBF &9J9(6IJM90J;>?(

%3! ; 63*, .)*!*6! - . * . ! ; 622+ ; *=63/! - . * . !+A*, . ; *=63!=)! . ! ; ,+ . *=C+/! .3 .2<*=; ! : ,6 ; +))! 8 0+,+/! =3)*+ . - ! 67! ? . *0+,=3?! ,+ . - =2<! +A> : 6)+ - !=376, (.*=63/! (+ .3=3?742!=376, (.*=63!=)!=37+, ,+ - !4)> =3?! .2?6,=*0 ( )B! #0+! +A*, . ; *=63! =*)+27! 5+ ; 6 (+)! : . ,*! 67! *0+! ;6 (: 6)+,L)! : ,6 ; +))! .)!)0+!= - +3*=7=+)! .3 - != (: 2+ (+3*)!)4=*> .52+! .2?6,=*0 ( )/!)+2+ ; *)! 8 0=; 0! - . * . !)*,+ . ( )! *6! =3 : 4*/! .3 - ! - +*+, (=3+)! 06 8 ! *6! : ,+)+3*! *0+! 64* : 4*! *6! *0+! - . * . ! : ,6 ; +)>)6,)B! # 8 6!= ((+ - = .*+! (+*06 -)!=3!3+* 8 6,9+ - ! (4)=; ! : +,76,> (.3 ; +)! . ,+! *06)+! 67! : ,6 ; +))=3?! *0+! .4 - =6!)=?3 .2!=*)+27! C= . ! - = ?=* .2!)=?3 .2! : ,6 ; +))=3?!PN"gQ! .3 - !+A*, . ; *=3?!=376, (.*=63! .564*!*0+!*, .3) (=))=63!*0,64?0! . !*= (+>?+6?, . : 0<! .3 .2<)=)B!

K<F<1 M+*%79":*+;79"')$?4::*;+"*

J 3+!67!*0+! (+*06 -)!76,!+A*, . ; *=3?! - . * . !7,6 (!)643 -!=)!N"gB! ')!6 : : 6)+ - !*6! (+* . - . * . ! ; 622+ ; *=63/!N"g!=)! .3! .2?6,=*0 (=; ! : ,6 ; +))!*0 . *! - + (.3 -)! ; 63)= - +, . *=63!67! . !C . ,=+*<!67!7 . ; *6,)K! *0+,+76,+/!=*!=)! . ! ; ,+ . *=C+! : , . ; *=; +B!N"g!*+ ; 03=D4+)! ; .3!+A> *, . ; *! C+,*=; .2! .3 - ! 06,=@63* .2!) : + ; *, .2! .3 - ! =3*+3)=*<! =376,> (.*=63!TWXVB!Z,6 (!*0+)+/! 8 +! ; .3! - + - 4 ; +! (4)=; .2! - +) ; ,= : > *=63)!7,6 (!*0+!)643 -/!)4 ; 0! .)!=3C . ,= .3 ; +!=3! : =* ; 0/!264 - 3+))/! - 4, . *=63/! .3 - !) : . *= .2! : 6)=*=63B! F)=3?! (. ; 0=3+>2+ . ,3=3?! *+ ; 03=D4+)/! 8 +! ; .3! +A*, . ; *! (4)=; .2!7+ . *4,+)/!)4 ; 0! .)! 5+ . *! * , . ; 9=3?/! ; 0,6 (. ?, . ( )/! .3 - !=3)*,4 (+3*! - +*+ ; *=63! .3 - !)+ : > . , . *=63!TWUVB! I =*0!*0=)!=376, (.*=63! . *!0 .3 -/!*0+! ; 6 (: 6)+,! ; .3! = (: 2+ (+3*! . !)<)*+ (! *0 . */! +B?B/! 2=)*+3)! *6! : =* ; 0! 6,!

melodic content and responds accordingly, or can synchronize events according to note onsets.

5.4 Time-geography approach

Time and space are intertwined in networked music performances. Time-geography [22] is an investigation into spatiotemporal processes that tell us about "potential encounters between agents" [23] and how those agents are constrained by space and time: e.g., by capability, coupling, and authority [22], where capability is the constraints dictated by the system in which the agent exists, coupling is being together in a place and time, and authority tells us about the forces that the agent is subject to. With these topologies, we expose the interconnected nature of the performers with the musical environment including the transmission network, where time-geography's "two main tenets are that time and space are seen as resources and that the constraints which operate on human beings particularly in the physical environment, are the primary dictates of human experience" [24]. A time-geography approach detects how agents are moving through time and space, and "distinguishes between fixed and flexible activities based on their degree of pliability in space and time" [25]. By combining telemetric analysis with musical feature analysis, time-geography techniques can tell us about under what limitations the system is operating: e.g., if a pattern experiences deviation due to congestion occurring over the network, which creates ripple effects for each agent's response time and thereby influences the musicians' ability to perform.

5.5 Pattern recognition

Where situations can be described as abstract logical problems, machine-learning algorithms are better than humans at detecting patterns. Automated pattern recognition techniques can use training data to detect relationships to existing known sequences, or they use the sequence itself to describe relationships within that sequence as the time series occurs. Pattern recognition implies having a pattern to recognize. For example, in networked music performances, we would simply train a recurrent neural network, such as long short-term memory or bidirectional recurrent neural networks [17] depending on our requirements, with initial telemetric data giving a base reading of the environment before a performance begins. Once the neural network has been trained, we will output data that can be folded back into the score.

5.6 Sequence deviation

Statistical reporting of deviations from a sequence, where anomalies and detected and measured, can be applied to both high-level agent movements in time, and within the agent itself when investigation change in pitch or harmonic sequences. In networked music performances, this determines when musical agents are subject to variation, and can be applied in scoring situations to detect both the creative decisions made by musicians and evaluate transmission performance within the network.

5.7 Sequence prediction

To add yet more layers of interpretation to her application, the composer may choose to predict the next events in a given time-series. Where the objective of a predictive model is to estimate unknown variables for two-dimensional sequences, such as pitch prediction, note onsets, envelopes, and network telemetrics, statistical forecasting algorithms are most useful where, at least for short-term forecasting, machine-learning algorithms offer little benefit at the expense of complexity [26].

5.8 Composite models

Integrating multiple algorithms and processes allows the detection and generation of multiple types and layers of sequences, allowing a more-sophisticated nonisomorphic approach to working with digital interfaces. When a massive data cloud can be processed, this exposes information about music as it happens and delineates the spatiotemporal relationships between remote participants. Data analyses give the composer and musicians insight towards both creating an informed performance environment and considering future potential opportunities.

6. *F NOT F*: ANALYSIS OF A NETWORK-AWARE SCORE

F not F (2019) is a continuous research project that puts into practice the principles introduced in this study: i.e., using machine-learning and statistical techniques to develop a responsive network-aware score interface. By analyzing the performance conditions, the score reacts to time-based pattern synchronization where the intent is to react to trajectories and thresholds that must be met before making certain decisions. To achieve this, a series of musical agents are deployed, where musical patterns that are distinguishable by the machine are analyzed.

The title of *F not F* refers to Nevejan's work on presence, where participants share or do not share time, space, and action. They may be "here" or "not here," and they may be participating "now" or "not now." That is, perhaps the musicians are responding to each other in real time or they are performing alongside a recording [5]. The title *F not F* suggests the mathematical concept of a function where input relates to an output.

6.1 Version 1

The first iteration of *F not F* was a custom Python program and a fixed notated score created for a live musical experiment between two pianists located in Ghent, Belgium, and Rotterdam, The Netherlands, which was presented at the Orpheus Institute's March 2019 conference on Simulation and Computer Experimentation in Music and Sound Art. During the performance, the pianist's sound was continuously analyzed by musical feature extraction methods using the Aubio[3] and Librosa[4] libraries. The program ingested real-time data that was sent to a computer vision program built on TensorFlow[5] and Keras[6] with the aim to determine what musical phrase was currently being performed. The program had been trained prior with 16 musical phrases, recorded hundreds of times in a variety of conditions, and translated into chromagram images (Figure 2). The chromagram was selected over the spectrograph or other feature- visualization methods after the testing of all of Librosa's options determined that the chromagram was the most easily recognized image by the computer vision software.

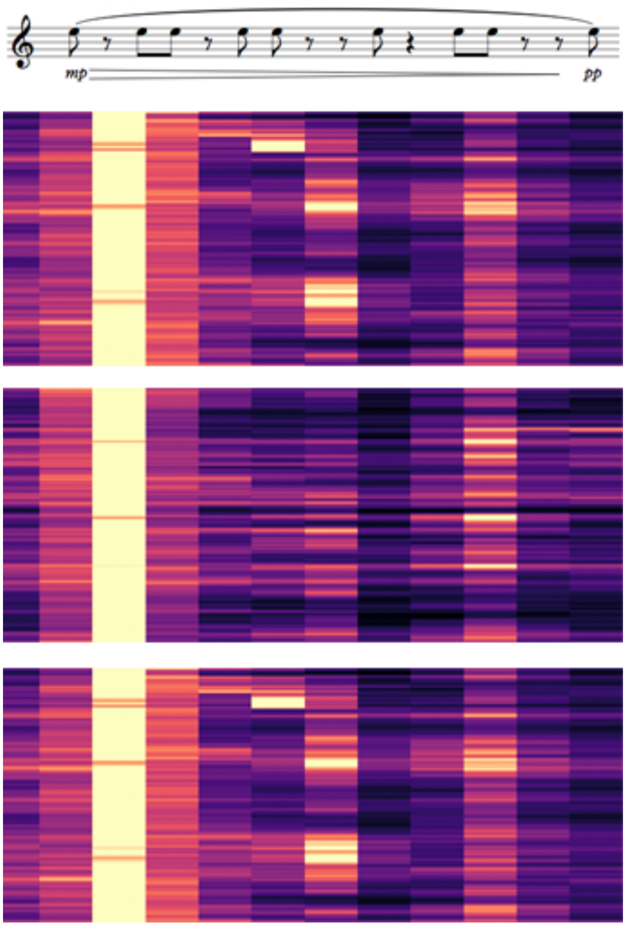

Figure 2. A musical phrase in *F not F* v. 1 and the chromagram representation of three recordings of that phrase.

These images served as training data to the program's machine-learning computer vision system. The performance consisted of the musicians playing a phrase a predetermined number of times, and then moving on to the next phrase. The discrete input of each pianist's mono-summed signal was recorded in real time by the program,

[3] Aubio is a C library and Python interface for the extraction of annotations from audio signals (see https://aubio.org).

[4] Librosa is a Python package for music and audio analysis for the purposes of music information retrieval (see https://librosa.github.io).

[5] TensorFlow is a platform for developing and training machine-learning models (see https://tensorflow.org).

[6] Keras is a high-level neural network Python API (see https://keras.io).

saved in 5-second wav files, one for each channel, and then immediately translated into chromagram images. The image was sent to the machine-learning program where the most-likely match was determined. This process took a few milliseconds. Once the program recognized a phrase, it triggered playback of samples from a bank selected according to the detected phrase. In all, a <6-second delay was experienced for the program to ingest and return the result. This delay informed the compositional structure where the musicians were instructed to repeat the musical patterns several times.

6.2 Version 2

F not F is a necessarily simplified implementation of the technical ideas discussed here; i.e., it is a constant work in development as the sophistication of tools improves. Machine learning is currently prone to error and ambiguity must be anticipated. Consequently, *F not F* is designed to embrace machine ambiguity while being structured overall for development of compositional narratives. Version 2 allows for a greater number of musicians and a variety of instruments to engage with musical agents. While version 1 was a static instructional score, a real-time score is generated for version 2, which can be viewed on a tablet connected to a central computer over a local or remote network. Like in the first version, an agent is a predefined microscore containing a distinct musical pattern where the rhythm and pitches can be easily detected by a machine. In version 2, agents become more sophisticated: i.e., a software program listens to each musician independently for the pattern and applies an interactive musical response. Table 1 lists events that are sent to the score once analysis has returned a result.

Analysis result	Action
A pattern has been partially detected	Trigger array of detected pattern string with threshold float value
A complete pattern has been detected	Trigger detected pattern string
The degree of synchronicity between any two data streams	Continuous stream of sync percentage (float array)
The degree of synchronicity between all streams (global synchronicity)	Continuous stream of sync percentage (float)
Telemetric data	Continuous stream of latency, packet loss, etc., to allow for synchronization calculations (e.g., mixed float, integer array)

Table 1. Analysis results and corresponding events

Version 2 of *F not F* is structured as a series of semi-notated precomposed instructions. For each series, musicians are presented with a series of notes. Figure 3 shows three examples of a note series that correspond to sections of a harmonic series, quantized to the semitone.

The musicians are instructed to improvise on the provided note series, playing notes in any order at a specified tempo and dynamic, but with deviating rhythms and timbres of their own choosing. For example, a resulting rhythm might sound something like the two rhythmic patterns shown in Figure 4.

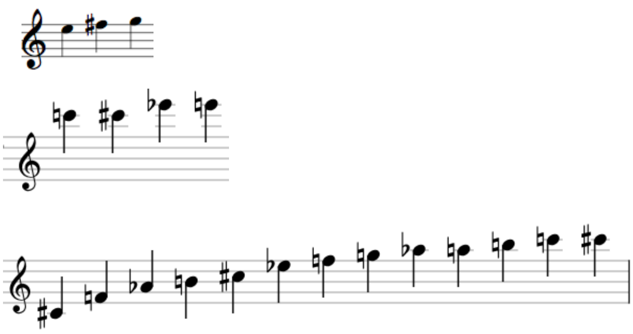

Figure 3. Three examples of a series: one for each musical agent.
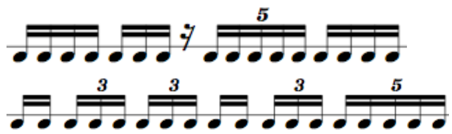
Figure 4. Two musician-determined rhythmic pattern examples.

From this improvisation, all instruments in the ensemble combine to create a semi-chaotic timbral and rhythmic effect that continues for an indefinite period of time. The machine listens for a pattern it recognizes, which initially will be a partial pattern. Once this event occurs, the detected pattern becomes the focus: i.e., the "musical agent" that matches that pattern is initialized and the agent pattern begins to emerge in the score. As illustrated in Figure 5, the musicians begin to interleave the agent pattern (top sequence) with the chaos series (highlighted lower sequence). The score encourages the musicians to gradually play the agent pattern gradually more and more by flickering between the two layers, while the machine increases its reading of partial-pattern detection length.
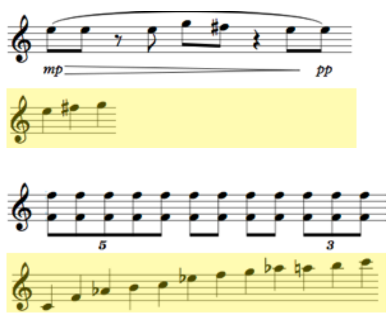
Figure 5. Two examples of an agent pattern that would be animated and interlaced with its corresponding chaos series.

The musicians should now be starting to play the agent pattern more than not, which allows the machine to match complete instances of the agent pattern and calculate the level of horizontal rhythmic synchronicity between musicians, reading telemetric data to calculate for latency. Once synchronicity crosses a determined threshold, the next series is triggered: i.e., the previous agent pattern disappears and a new chaos pattern is shown to the musicians. This cycle can continue as long as the participants and composer wish, by supplying more or fewer patterns until the machine has completed its tasks.

An agent may also have a particular set of parameters that determine its musicality: e.g., it may respond to selected types of data streams or unique triggers specific to that agent to generate correlated effects using granular synthesis or other processing effects. In *F not F*, an agent may be responsive to the current global sync value in real time and react as outlined in Table 2.

NOT IN SYNC (<20%)	**WITHIN THRESHOLD (>20% to <80%)**	**IN SYNC (>90%)**
No effects	Clicks, noise bursts, sine tones	Agent sample triggers
	Increasing granular synthesis level	Granular synthesis immersion
		Countdown to next series

Table 2. Triggering electronic effects in *F not F*

When the musicians are not in sync, there are no effects. This decision was made so they can more easily seek synchronicity with each other. As their sync increases, so do the effects and granular synthesis levels. The intent of increasing and decreasing the intensity of electronic effects is to offer the musician the musical choice of whether she wishes to move towards or away from sync, depending on her musical intentions in accordance with the ensemble.

7. FUTURE WORK

The long-term goal of this research study is to develop advanced methods for detecting how musical agents are created and transmitted between networks. Learning how they interact and modified and transformed, whether by musician or machine, guides us towards understanding how participants experience and react to being in-time and in-rhythm with each other when performing remotely over a network. While being a disruptive factor in music, latency can be used for positive gains with machine interaction. The unavoidable delay can be well-utilized, where the transmission wait can be used to analyzed and generate performative interfaces for musical developments. Developing complex network-aware systems demands the intersection of real-time machine protocols with human-level research on mediated presence technology and theory and computer music studies. Collecting data from as many sources as possible at the time of creation and processing that data with algorithmic approaches allows us to create sophisticated sound processing applications and score creation for distance-aware composition structures while building strong musical relationships and transferring the vital information that musicians need when performing remotely. When working with sophisticated graphical interfaces, the transmission latency could be highly beneficial, e.g., when developing complex, immersive works using virtual and augmented reality tools.

Most machine-learning and statistical technologies to date are focused on genre classification or harmonic analysis. A real-time library for detecting and comparing sound features such as timbre, note onset and decay, and agent detection and comparison, e.g., would allow the composer to consider greater score and compositional complexities. As detection and analysis tools become more sophisticated, such as the real-time application of deep learning tools to recognize activities [27], composers will have the ability to model and detect musical features at fine resolution and great speed at their fingertips [28]. In addition, structural and decision-making events can be analyzed and cocreated by musicians and machines. In step with advances in analysis, innovations in digital scoring technologies increase the possibilities of representing this information with highly sophisticated responsive musical agents, which leads to more meaningful encounters between remote musicians and computing systems. As greater amounts of data become available, human perception tends towards perceiving greater meaning; i.e., we attribute a machine's complex recognizable patterns to mechanisms imbued with agency.

Ultimately, there are undiscovered depths of exploration in relation to performance and the machine, and our continued curiosity will certainly uncover many more aesthetic and technical applications. This research works towards the creation of a functional library of performance and scoring tools, while growing the portfolio of networked music performance works by this composer. She hopes to entice others to approach networked music performance as a source for novel and intriguing aesthetic opportunities.

8. REFERENCES

[1] C. Nevejan, Urban reflection: On the design of diverse engagement in the networking city of Amsterdam. [Online] Available: https://nevejan.org

[2] C. Nevejan and F. Brazier, "Granularity in Reciprocity," *AI & Society*, vol. 27, no. 1, 2012, pp. 129–147.

[3] P. Oliveros, S. Weaver, M. Dresser, J. Pitcher, J. Braasch, and C. Chafe, "Telematic Music: Six Perspectives," *Leonardo Music Journal*, vol. 19, no. 1, 2009, pp. 95–96.

[4] S. P. Gill, "Tacit Engagement: Betwixt and In-between," in *Tacit Engagement*, Springer Nature, 2015, pp. 127–160.

[5] C. Nevejan, *Presence and the Design of Trust*. Dissertation, University of Amsterdam, 2007.

[6] J. Croft, "Theses on Liveness," *Organised Sound*, vol. 12, no. 1, 2007, pp. 59–66.

[7] R.R. Smith, Composer Ryan Ross Smith lectures on his 'Study no. 47' at the New Music Gathering. [Online] Available: https://www.youtube.com/watch?v=p66Quh8zDl8

[8] S. Emmerson, M. Baalman, and O. Brandtsegg, "Instrumentality, Perception and Listening in Crossadaptive Performance," in *ICLI 2018 – International Conference on Live Interfaces (Porto)*, June 14–16, 2018, pp. 86-95.

[9] G. Hajdu, "Automatic Composition and Notation in Network Music Environments," *Proceedings of Sound and Music Computing Conference, SMC 2006 (Marseilles)*, 18–20 May, 2006.

[10] Quintet.net. [Online] Available: https://quintetnet.hfmt-hamburg.de/wiki/projects/quintetnet/Quintetnet.html

[11] I. Whalley, "GNMISS: A Scoring System for Internet2 Electroacoustic Music," *Organised Sound*, vol. 19, no. 3, 2014, pp. 244–259.

[12] S. James, B. Carey, X. Fu, and G. Hajdu, "Establishing Connectivity Between the Existing Networked Music Notation Packages Quintet.net, Decibel ScorePlayer and MaxScore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation, TENOR'17 (A Coruña)*, 2017, p. 13.

[13] E. Cayko, *Rhythmic Topologies and the Manifold Nature of Network Music Performance*. Dissertation, University of Calgary, 2016.

[14] G. Hajdu and N. Didkovsky, "On the Evolution of Music Notation in Network Music Environments," *Contemporary Music Review*, vol. 28, no. 4–5, 2009, pp. 395–407.

[15] R. S. Hatten, *A Theory of Virtual Agency for Western Art Music*. Indiana University Press, 2018.

[16] G. Peeters, *A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project*. IRCAM, 2004.

[17] N. Farsad and A. Goldsmith, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, 2018, pp. 5663–5678.

[18] T.L. Li, A.B. Chan, and A.H. Chun, "Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network," *Genre*, vol. 10, 2010, pp. 1x1.

[19] J.A. Burgoyne, I. Fujinaga, and J. S. Downie, "Music Information Retrieval," in S. Schreibman, R. Siemens, and J. Unsworth (Eds.), *A New Companion to Digital Humanities*, John Wiley & Sons, Ltd, 2015, pp. 213–228.

[20] M. Muller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal Processing for Music Analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, 2011, pp. 1088–1110.

[21] B. Srebrenik, Machine learning and music classification: A content-based filtering approach. [Online] Available: https://towardsdatascience.com/machine-learning-and-music-classification-a-content-based-filtering-approach-f2c4eb13bade

[22] T. Hägerstraand, "What About People in Regional Science?," *Papers in Regional Science*, vol. 24, no. 1, 1970, pp. 7–24.

[23] S. Winter and Z.-C. Yin, "The Elements of Probabilistic Time Geography," *GeoInformatica*, vol. 15, no. 3, 2011, pp. 417–434.

[24] N. J. Thrift, "An Introduction to Time-Geography," In *Geo Abstracts*. University of East Anglia, 1977.

[25] H. J. Miller, "A Measurement Theory for Time Geography," *Geographical Analysis*, vol. 37, no. 1, 2005, pp. 17–45.

[26] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward," *PLoS One*, vol. 13, no. 3, 2018, p. e0194889.

[27] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep Learning for Sensor-based Activity Recognition: A Survey," *Pattern Recognition Letters*, vol. 119, 2019, pp. 3–11.

[28] R. Fiebrink, "Machine Learning as Meta-Instrument: Human-Machine Partnerships Shaping Expressive Instrumental Creation," in Musical Instruments in the 21st Century, T. Bovermann, A. de Campo, H. Egermann, S.-I. Hardjowirogo, and S. Weinzierl, Eds. Singapore: Springer Singapore, 2017, pp. 137–151.