Ambient Data Monitoring w/Generative Music Systems using EC & ML Techniques.

Stephen Roddy

CONNECT Research Centre,
Trinity College Dublin,
Dublin, Ireland
roddyst@tcd.ie

Abstract. This is a position paper which describes work in progress to develop an AI/ML driven auditory ambient information system which incorporates generative music techniques and considers some of the factors involved the design and development of a system of this nature. The system is intended to represent and communicate information about cryptocurrency markets to a user. The generative music system uses evolutionary computing and machine learning techniques, which are driven by the original cryptocurrency data, to create musical information. This information is then mapped to synthesis parameters. The implementation described is web based running client-side in the browser. The paper describes the relevant background literature before presenting a generalised model for a system of this nature before discussing aspects of the implementation in its current state. The paper closes with a discussion of the system and future work.

Keywords: Ambient Information System, Variational Autoencoder, Genetic Algorithm, Auditory Display, Sonic Information Design, Web Technologies.

1 Background and Introduction

This section deals with the question of why the ambient monitoring of cryptocurrency market data is relevant and why a data-driven, generative music approach is suited to this task. It introduces and explores the diverse topics of cryptocurrency markets, sonic information design the relationship between generative music systems and ambient music as well as the concept of the ambient information system. It ties together a number of strings from these topics to present a two part research questions that is then further explored in section two of this paper.

1.1 Cryptocurrency Markets

A cryptocurrency is essentially a cryptographically secured digital currency built on a blockchain. Originally introduced by the mysterious Satoshi Nakmoto in his White Paper the Bitcoin cryptocurrency [1], a blockchain is a distributed ledger that uses a cryptographic technique called a hash function to allow information to be shared across a large network without actually being copied. In essence each block in a blockchain is comprised of its own information as well as a hash of the information contained in the previous block. This makes it difficult to tamper with as changing a single block would require all following blocks in the sequence to be changed also. With this technology distributed ledgers can be generated by publicly sharing a blockchain across all nodes in a given network and introducing validation procedures for the encoding of a new block of information. Cryptocurrencies are basically tokens that are encoded on such a blockchain network.

Thanks to a much discussed bubble the cryptocurrency market value hit 566.3 billion towards the end of 2017 up from 1.3b over the five year period from May 2013 [2]. More recent moves towards regulation of what has become a highly speculative market saw the market value tumble to 335 billion in February and back to 429.6 b as of writing [2]. The cryptocurrency market is highly volatile. It has also attracted a lot of interest and investment from outside of the traditional financial investment circles [3]. Large numbers of young and amateur investors have flocked to the cyrptocurrency market many of whom hold a primary occupation in areas unrelated to finance. This places them at an obvious disadvantage to those who are investing in a professional capacity and are in a position to spend much of their day monitoring and working with cryptocurrency markets directly [4]. Recent FinTech relevant advances in the fields of Artificial Intelligence, Machine Learning and Sonic Information Design suggest that this gap might be addressed through the application of techniques and technologies which can help to level the playing field between the professional and amateur cryptocurrency investor.

1.2 Sonic Information Design

Sonic information design refers to the application of design research, as defined by Faste and Faste [5], to *sonification*, an auditory display technique in which data is systematically mapped to non-speech sound for the purposes of representation or communication [6]. In this context, design research is taken to involve “the study of design and the process of knowledge production that occurs through the act of design” and can be roughly defined as “the investigation of knowledge through purposeful design” [5]. Where visual information design is generally concerned with the presentation of information in a manner that can be effectively and efficiently understood [7] sonic information design "pays particular attention to user experience including physical, cognitive, emotional, and aesthetic issues; the relationship between form, function, and content; and emerging concepts such as fun, playfulness and design futures" [8].

Auditory display researchers are divided over the topic of music and its relationship to auditory display and sonification. Voices on one side argue that it is not necessarily useful or meaningful to distinguish between musical and non-musical sonifications [9] and those on the other side argue that music and sonification are inherently different practices, sonification being scientific and music being artistic [10], that should not be conflated if the field wishes to gain the acceptance of the wider scientific community [11]. Supper [12] recognizes this as an essential tension in the field and something of a limiting factor to its further development. However, within the context of sonic information design this essential tension can be overcome as auditory display and sonificaiton move away from purely scientific or artistic contexts towards a design based approach that can integrate aspects of both fields as required. In this context it has been argued that the bulk of research in the field of auditory display can generally be associated with and understood through the lens of the Second Wave of HCI [13] as defined by Bannon [14] and Bødker [15]. However Sonic Information Design has emerged in the context of the embodied and aesthetic turns in auditory display [16][17] aligning more closely with the methods and techniques of Third Wave HCI which is increasingly focused on meaning-making and opens up a space for more creative and imaginatively motivated approaches to auditory display and sonification. This paper is focused on data-driven generative music systems which can be used for the ambient monitoring of highly volatile cryptocurrency market data.

1.3 Generative Music Systems and the Birth of Ambient Music

Algorithmic approaches to music composition, the use of formal rule sets in the creation of music, have a long history in Western culture being explored by thinkers ranging from Pythagoras to Mozart and more recently Lejaren Hiller and John Cage [18][19]. The term "Generative Music" entered circulation when Brian Eno [20] coined it in relation to his compositional activities in 1995 with the Koan system (superseded in 2014 by Intermorphic's Wotja). Eno thought of generative music as system-propagated music that is in a state of constant flux. He quickly became a pioneer and advocate of generative music practices producing a wealth of generative music compositions across his career [21] and more recently collaborating on a series of mobile apps for generative and ambient music with Peter Chilvers [22]. While the term rose to prominence in the 1990s Eno recognized that like algorithmic music, generative music had existed in some form or other since at least the invention of the Wind Chime [23]. In fact Eno himself had been making music with generative systems since his release of Discreet Music in 1975 [24] and further explored generative processes in Ambient 1: Music for Airports [25]. Discrete Music involved a simple generative music system which used a delay line to reconfigure the collation of dual melodic loops giving rise to seemingly endless harmonic possibilities in a musical form that was "part of the ambience of the environment". This approach is somewhat similar to approaches adopted by Steve Reich and most especially his 1965 piece *It's Gonna Rain* [26]. This experiment in generative music was also Eno's first step towards Ambient Music which would be more fully realized on Ambient 1: Music for Airports. A similar generative system was adopted for this record creating a new instrumental musical form which is intended to sit in the background on the periphery of ones awareness creating an atmosphere or environment rather than defining a strict musical narrative. This relationship between generative musical practices and the birth of the ambient music genre in Eno's work of key importance to the system described in this paper.

1.4 Ambient Information Systems

Ambient information systems are aesthetically pleasing displays of information which sit on the periphery of a user's attention and support them in monitoring of information that is of less importance than their primary work task. While they are designed for a variety of modalities, this paper is concerned with auditory ambient information systems. Pousman and Stasko [27] associate ambient computing with the concept of Ubiquitous Computing and more specifically Weiser and Brown's [28] concept of "calm computing." These points echo Eno's intent for ambient music to "induce calm and a space to think" [29]. As such generative music systems might be particularly well suited to applications in the context of ambient information systems as generative music techniques have been integral to the birth and development of ambient music composition which, like the ambient information system, is concerned with aesthetic presentation in a non-intrusive manner. With the introduction of a sonic information design component, generative music systems might be mapped to some data in order to present information to a listener in the context of an auditory ambient information system.

Such systems may be of particular relevance for the amateur cryptocurrency investor who is primarily employed in some occupation outside of financial investment. It allows them to focus on the task at hand while retaining access to information throughout the day about their investments in volatile cryptocurrency markets. The volatility of these markets means that close monitoring is required on the part of investors and traders who are interested in the short-term performance of these markets. Systems of this nature might also be of use to professional who needs to monitor a large amount of market data or information and can off-set some of that load with the help of an ambient information system.

This in turn leads to the central questions of this paper:

1. How can we use data-driven generative music strategies to create an ambient information system for cryptocurrency market monitoring?

2. What tools are available for leveraging generative music systems for the design and development of data-driven ambient information systems?

A data-driven generative music system for ambient monitoring needs to accomplish a number of tasks.

1. Accurately represent and communicate the Data.
2. Be easily Interpretable by the user.
3. Remain Non-Intrusive while the user is otherwise engaged.

A data-driven generative music system involves the parameterization of a generative music system such that meaningful changes in the original data source are made manifest in the music generated. By adhering to the aesthetic principles of ambient music we can ensure that a data-driven generative music processes can be deployed in the context of an auditory ambient information system, remaining on the edge of the listeners awareness but accessible nonetheless when needed. Recourse can be made to literature on auditory display design, sound in HCI, cognitive science and the bourgeoning field of sonic information design to determine strategies by which the system can accurately communicate the data and remain easily interpretable to a user.

1.5 The State of the Art

The state of the art has moved on quite a bit since the early days of generative music as the tools and techniques for creating generative music systems have become increasingly sophisticated and advanced. Papadopoulos and Wiggins [30] present a survey of AI techniques which have been used for algorithmic music composition. They review the application of mathematical models (e.g Markov chains), knowledge based systems (symbolic rule based systems), grammars like Cope's [31][32][33] EMI (or Emmy) system, evolutionary methods (e.g. genetic algorithms and cellular automata) and systems which learn (e.g. neural networks and hybrid systems which combine multiple techniques. Collins [21] presents summary of the many programs and programming languages available for the creation of generative music in a real-time performative and compositional context and Briot et al. [34] present an in depth survey of deep learning approaches and their applicability to generative music composition. The field has seen much development form a theoretical point of view also. Miranda's [35] *Evolutionary Computer Music*, comprised of chapters authored by key thinkers and practitioners in the field, has contributed to the systematization of thinking in regards to evolutionary methods in generative music while Boden's [36][37] research on the relationship between creativity and computing has influenced the design of a number of generative music systems [38] [39] [40] [41] [42][43]. While advances have been taking place in the field of generative music much greater advances have been taking place in computing more generally with the emergence and maturation of the Internet and recent developments in IoT, cloud, mobile and smart technologies [44]. These technologies have reshaped Western culture and defined a new set of practices around how we consume and interpret information and create and listen to music [13][44]. This cultural shift towards an online and networked approach to information consumption and music listening needs to be addressed and accounted for in the design of a system which is intended to present key information to a listener through music. Services and functionalities, from word processing to video editing, which could previously only be provided by desktop applications are moving online.

2.1 Overview of the System

Figure 1 presents a general model describing how data-driven generative music techniques can be employed in an ambient information system for monitoring cryptocurrency market data. The implementation described in this paper is currently in active development and this process is guided by the model presented in figure 1. The first layer acquires data through a relevant API. This data is then processed so that it is in a useful range which can be mapped to control and drive parameters across the system. There are two distinct aspects to the generative system. The first uses evolutionary computing methods to generate musical information while the second employs machine-learning techniques. Data is mapped to control how these components generate musical information before that information is submitted to the synthesis layer. In the synthesis layer musical information is mapped to synthesis parameters and data can be further mapped to control synthesis parameters. In the post production layer the audio signals created in the synthesis layer are processed further and this processing is controlled by the data also. Each of these processes is underpinned by relevant libraries and frameworks and each layer in the model can be interacted with through a front end framework. This front-end framework can, for example, allow the user to choose between evolutionary computing approaches and machine learning approaches in the generative layer.
Generally smart technologies data, spatial media data, network traffic and IoT data are represented visually through data dashboards. An auditory ambient information system can be used to compliment this approach by allowing users to focus on the core data streams of relevance to a specific, while monitoring less vital information in the background. Key to this approach is the concept of the "target state". This was a critical design concept for the sonic representation of data to emerge in research conducted with the Pervasive Nation IoT Network [57]. A target state is simply a state of interest to a user that is indicated by the some predefined number of monitored data-streams falling into specific pre-defined ranges. These target states in the data can be paired with specific sonic patterns, or target sound, indicating to a user that the state has been reached. The task of the system is to inform the listener when one of these target states has been reached and outside of those times to represent information to the listener as a measure of distance from one of those target states. This allows the user to gauge their "distance" from a target state on the basis of how similar a given sonic pattern is to a target sound. This approach also supports the listener in determining whether the data is trending towards or away form the target

state. The following sections describe an implementation of the model and discuss the functionality of a number of the layers in greater detail.

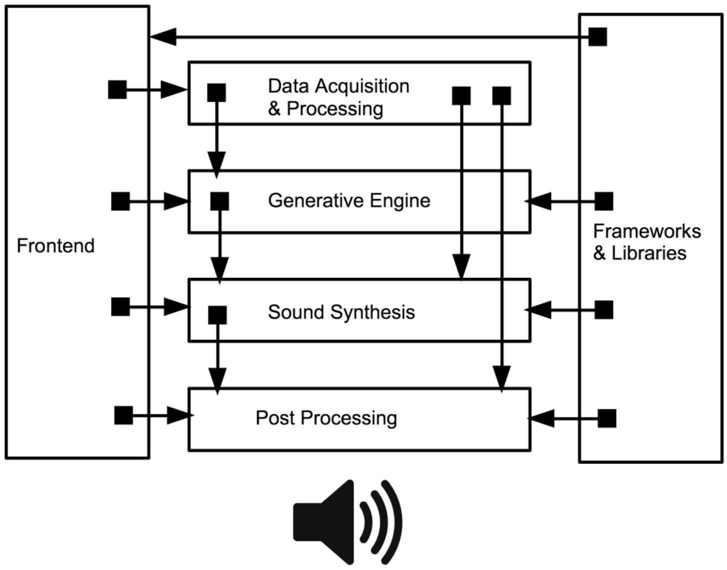

Fig. 1 Data-driven Generative Ambient Information System Model

Web app architectures which carry out the bulk of their processing client-side, in the browser, are becoming increasingly relevant as we move towards a context-aware, ambient internet serving as a pervasive and ubiquitous platform for powerful apps. In the era of the Ambient Computing, Progressive Web Apps (PWAs) will offer the functionality of native mobile apps from standard web pages [45] while distributed apps or D-apps will offer similar functionality while running on specific blockchains [46]. The standard client-server architectures prevalent prior to the era of mobile and smart computing are expected to become less and less relevant as the internet becomes more widely distributed and decentralized [46] [47] as functionalities that were previously driven by server-side architecture are increasingly running in the browser on client-side architectures driven by powerful JavaScript frameworks [48]. Recent developments in web technologies, discussed later in this paper, have made the deployment of client-side evolutionary computing, machine learning and sound synthesis methodologies feasible.

The implementation discussed below utilizes the core web technologies: HTML, CSS and JavaScript. The development of dynamic and single-page applications (SPA) have been supported by JavaScript frameworks like Node.js, Angular.js, Backbone.js for many years (2009, 2009 and 2010 respectively). Node.js has particularly encouraged the development and proliferation of JavaScript libraries as it executes JavaScript, a traditionally client-side scripting language, on the server [49]. More recently a wide range of JavaScript libraries and frameworks have appeared which are leveraging the WebGl [50] and Web Audio APIs [51] to extend the traditional capabilities of JavaScript arguably bringing it closer to the status of a platform rather than just a language. This trend is set to continue as at present the W3C are adding a set of components, termed Web Components [53], to the HTML and DOM specifications. Web components has the potential to revolutioise web app development especially in the context of progressive web apps and Google's Polymer library, written in JavaScript is already allowing developers to access and implement some of these new features and functionalities by creating their own custom elements [53]. APIs (application programming interfaces) are critical components driving much of the functionality of todays Internet. An API is essentially a set of tools that allow one application to access and leverage data and functionalities from another application [54]. For example the Google Maps API allows developers to embed a fully functional Google map in their own applications. A general shift in web development practices and business models for online companies towards SaaS (Software as a Service), B2B (Business to Business) and platform models has driven the API economy, the exchange of value between tech companies on the basis of their APIs [54]. In January of 2018 the ProgrammableWeb which maintains a database of core APIs topped 19,000 APIs up from 17,000 the previous January and 121 in January 2006. Accounting for the fact that 90% of APIs are private and never listed at ProgrammableWeb the API economy is booming [55]. This API driven web ecosystem is valuable for the development of ambient information systems for Cryptocurrency monitoring as many of the online platforms for buying, selling and monitoring currencies offer fully featured API's which support developers in building their own monitoring tools. The following sections explore an implementation of the model describing the functionality of some of the model layers in greater detail.

2.2 Data Acquisition and Processing

Data acquisition is managed in JavaScript using XMLHttpRequest (XHR) objects. A number of APIs offer up to date cryptocurrency information for trading. This application uses the Global Digital Asset Exchange (GDAX) API [56] and data is updated once every second. Incoming data streams are rescaled to represent a percentage of the all time high market cap for the currency represented. This data can then be mapped to modulate parameters in the generative, synthesis and post processing layers.

2.3 The Generative Engine

The generative layer of the model drives the generation of musical information that can then be mapped to synthesis parameters in the following layer. The generative engine in the system implementation described has two functionalities. It employs a genetic algorithm to generate musical information, and it also employs a variational autoencoder, a machine learning technique, to generate a latent space of musical information. These points are explored in further detail below.

2.3.1 Evolutionary Computing

When employing evolutionary techniques for the purposes of representing data using generative music the question of how best to paramaterise the algorithm or how to map the data to control the algorithm is of key importance. Historically when GAs have been used in generative music, fitness functions have been determined by the composer [30]. A composer will generate some music, listen to it and decide whether to keep it or dispose of it. For the purposes of representing and monitoring data we can map data to the fitness in the system described previously. In this case target states in the data can be represented by the target solutions for which GAs are solved.

In this way as the data tends towards a certain value the GA will tend towards a target solution, and the musical output will tend towards a specific motif. This dynamic can then be leveraged so that a value of interest in the data is represented by specific musical or sonic pattern. The ambient information system then describes the distance of the current data value from the target value through the "distance" between the current musical content heard and the target musical pattern.

There are many libraries for evolutionary computing listed on the JavaScript package manage npmjs.com and even more repositories on github featuring different libraries and frameworks. While the majority of these tend to focus on specific applications Genetic.js [58] casts as a general purpose tool. Genetic.js is a lightweight library of genetic and evolutionary algorithms that can be run in the browser. It contains a number of useful functions constituting some of the basic building blocks of evolutionary computing, e.g. functions for evaluating fitness and performing mutations and crossovers. The library can be used to build simple genetic algorithms or solve more complex operations such as curve fitting and phrase solving. [58]. In similar fashion the inveniemus.js library is intended for search and optimisation applications that includes genetic algorithms, particle swarm optimisation and differential evolution as well as a number of other metaheurstic algorithms. It can be run in the browser or server side with Node.js. JeneticS.js [59] is another similar library for creating genetic algorithms. evospace.js [60] and NodEO [61] are more heavily reliant on the support of server side architecture implemented in Node.js though nodEO can be converted for use in the browser. While all of these solutions are serviceable none of them are adapted to work with musical information straight out of the box. Furthermore some of these libraries are quite bulky and rather than loading an entire library and using up valuable resources while creating excess redundancy the choice was made to write genetic algorithms specifically tailored to the creation of melodic, harmonic and rhythmic content in the MIDI format. The design of this algorithm was guided and informed by the literature [35].

2.3.2 Machine Learning

Google's Tensorflow [62] is an open source Python framework for deploying machine learning across a varied range of platforms and devices. It has become an invaluable tool for machine learning research and development.

Tensorflow.js, [63] the successor to deeplearn.js, is a recently introduced (March 2018) WebGL accelerated, browser based JavaScript library for training and deploying ML models. It runs ML applications client-side in the browser. Tensorflow.js can import existing pretrained ML models to perform inference in the browser and it also supports the retraining of imported models as well as the creation of new models in the browser. Tensorflow.js represents an important step in the proliferation of ML technologies in an online context and is very good news for browser based generative music. ML methods have been successfully deployed in a generative music context. DeepMinds's WaveNet [64] is a deep generative model for audio waveforms like speech and music. It is a deep autoregressive network of dilated convolutions (CNN) which models sound on a sample by sample basis inspired by Pixel-RNN which was also developed by DeepMind. WaveNet has a number of shortcomings in the context of the generalized model presented here. It is trained on and outputs raw waveforms which conflicts with the models prescription that the generative engine output information that can then be mapped to sound after production. Furthermore it is difficult to map the data to control the output without interfering with the network at run time and there is currently no way of running WaveNet in a browser.

The Magenta Project's NSynth [65] employs a WaveNet-style autoencoder that conditions an autoregressive decoder on temporal codes learned from the raw audio waveform. The NSynth training data-set consisted 305,979 musical notes of 3 seconds length with a 1 second ring out phase. Each one had a unique timbre, and envelope. These were sampled from 1,006 unique commercial sample libraries across the standard MIDI piano pitch range (21-108) at five MIDI velocity levels (25, 50, 75, 100, 127). In essence NSynth network creates blends of characteristic timbral features from the input dataset. NSynth has been released as a MaxForLive device. The Magenta project has introduced some important advances in the application of machine learning techniques to art and music. In a web context Performance RNN [66] employs an LSTM-based neural network for generating polyphonic music.

The network was trained on the Yamaha e-Piano Competition dataset, comprised of MIDI data from roughly 1400 competition performances. Crucially this data contains velocity values and fluctuations in micro-timing which allow the network to model expressive dynamics. More recently (May 2018) they released Magenta.js [67] which includes implementations of a number of Magenta's music models to be run in the browser. The Melody RNN model applies language modelling to melody generation using an LSTM while ImprovRNN conditions an underlying chord scheme on melodies generated by MelodyRNN. DrumsRNN works in a similar fashion to MelodyRNN but with drums. Crucially these models are trained on and produce MIDI information rather than raw sound files so they are more compatible with the generalised model than WaveNet and NSynth. They provide useful tools for generating musical content. Of critical interest in the context of data-driven generative music however is MusicVAE [68]. This is a hierarchical recurrent variational autoencoder for learning latent spaces of musical features. Autoencoders differ from standard neural networks in that the input is the same as the output. As such they can be used as generative models. An autoencoder creates a compressed representation of features in a high dimensional data-space. The decoder can then recreate the original input from this lower dimensional representation called a latent space. A variational autoencoder also learns to model the original data with a probability distribution. Sampling from the latent space on the basis of this probability distribution allow for the reconstruction of attribute vectors which represent smoothly and continuous sequences of learned properties from the original data-set. In essence NSynth generates a latent space of timbre populated by different blends of characteristic timbral features from the input data-set. MusicVAE generates latent spaces of musical content, represented in MIDI where adjacent points in the space share similar but slightly different properties.

The potential here for data-driven generative music consists in mapping input data to control which points in the latent space are passed to the synthesis engine. Moreover because of the continuous attribute vectors of in the latent space increases in specific properties of the data can be mapped to increases in analogous or metaphorically related sonic properties.

Using a variational autoencoder, and MusicVAE more specifically, a target state in the data can be represented using a melodic pattern. The listener can then gauge how close or far the data is to the target state on the basis of how close or far the melody is from the target melody.

2.4 Sound Synthesis and Post Processing

There are a number of JavaScript libraries and frameworks for synthesizing audio in browser; many of them utilize the Web Audo API. Tone.js [51] is a framework built on top of the Web Audio API that provides a set of synthesis algorithms, effects, event scheduling tools and musical abstractions suited to creating interactive browser based music. Flocking.js [69] is another framework built on the Web Audio API which adopts a declarative approach to creative sound and music programming.

WebPD [70] is a runtime JavaScript environment that is also built on the Web Audio API and runs Pure Data patches in the browser. This may present an obvious advantage over the other libraries for Pure Data users. Another library of note is Gibber [71] which provides a live coding environment for audiovisual performance and composition. While not necessarily well suited to this particular context it nonetheless provides useful and interesting functionalities. The system described in this paper uses Tone.js. Tone.js allows for the control of low level synthesis parameters as well as higher level musical parameters. This makes it a good fit for use with the musical information output by the generative component of the system and also means that synthesis parameters can be more directly controlled by the incoming data stream. Tone.js also contains a number of effects functions that can be used for post processing. The data can be directly mapped to parameters including reverb, panning, pitch-shifting, granulation, convolution, chorusing, delays, filters and distortion. While these can be used in the traditional manner they are applied in standard studio production (e.g to achieve the best possible presentation of a sonic idea) they can also be used for communicative purposes. These effects routines can be used to create spectromorphological sound shapes [72] from the harmonic materials created by the system. Data can be mapped to control these sound shapes allowing for the representation of data across an additional sonic layer. Testing is currently being carried out to determine how effectively Tone.js's effects modules can be used to recreate some of the spectromorphological sound shapes described by Smalley [72].

3.1 Discussion

There are a number of novel aspects to this work. The concept of matching target states in the data to specific sonic representations is novel. Generally approaches in sonic information design are informed by parameter mapping sonification practices. This approach is focused on defining a systematic mapping strategy to describe how the data should be mapped to synthesis parameters. The approach described in this paper side steps that issue. There is of course a systematic mapping strategy involved, but the focus is not on the conscious definition or design of the mapping strategy. Instead much of the mapping strategy design is deferred to the genetic algorithm and variational autoencoder which produce musical information. The task then shifts from mapping data to synthesis parameters to mapping the musical outputs of these generative models to synthesis parameters. This may help to contribute to finding a solution for the mapping problem [73], which is a general open problem in the field of auditory display that asks how data should be mapped to sound in order to accurately represent it to a listener.

The target states approach also provides a novel solution for the problem of representing useful information a user. Approaches in the field of auditory display tend to involve mapping all of the data to sound in the hopes that some kind of high-level meaning emerges for a listener, even though this is rarely the case [73]. This approach allows for the most important aspects of the data to be represented. Monitoring the changes in a data set on the basis of distance from a target sonic patterns is also another novel aspect of this approach. It shifts focus away from the usual kinds of cognitive and listening skills employed in auditory display design, where value changes are often represented using pitch changes. This allows users instead to rely on their cognitive capacity for matching sonic patterns to interpret the data. Given the musical nature of the system in question it would be expected that musically inclined listeners would find it easier to interpret the data.

Recent research has advocated for the creative potential represented in latent spaces generated by variational autoencoders [74], and this was realised musically in Magenta's creation of MusicVAE [64]. This project represents an early adoption and application of these methods to problems in sonic information design. A shortcoming of the system is that the implementation of MusicVAE used in this project has been trained on the Yamaha e-Piano Competition dataset. In order to preserver the ambient aesthetic aimed for in the system it would make more sense to train MusicVAE on a data-set comprised of relevant ambient music compositions, future work will aim to address this shortcoming.

3.1.1 Data Dashboards and Future Work

At present the core functionality of the system has been implemented allowing the listener to monitor changes in cryptocurrency data streams through an ambient auditory information display. There is still a need to build a front-end that will allow the user to interact with the code and determine some of the parameters across the different levels of the system. With this is in place the usability of the system will be gauged through user centric evaluation. After this current plan is to incorporate the functionality into a broader data dashboard. Data dashboards aggregate important data and information for the purposes of monitoring and analysis providing a visual summary of critical information to aid users in a variety of tasks and decision-making processes [75]. They have become critical in the context of smart cities [76][77], network traffic and security monitoring [78][79] and asset management in business and financial contexts [79][80]. A number of researchers have recently begun to explore the limitations involved in the visual display of data to users in the context of data dashboards [81][82]. Incorporating the functionalities discussed here would allow for some of visual data to be offloaded to the ambient information system, thus recruiting the listeners' auditory cognitive abilities to support

Another plan for future work is to further adapt the system to create a generalized sound enabled data dashboard template in Angular.js. This template can then be adapted for application to problems in a number of related spaces including smart asset management, smart city monitoring and IoT network monitoring. One such application of the template will be the Pervasive Nation, a nationwide internet of things testbed operated by CONNECT. Earlier iterations of the implementation described here were written in Python and Csound for use with the Pervasive Nation network. It is planned to eventually adapt the finished template to the Pervasive Nation.

A further goal is to introduce the concept of musical interaction to the implementation. The system will be expanded to allow a user to perform a piece of music using their keyboard or a MIDI controller. This information will be captured and used to represent the target states for the genetic algorithm and variational autoencoder.

References

1. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. (2008)
2. CoinMarketCap.com, https://coinmarketcap.com/
3. Fry, J., Cheah, E.T.: Negative bubbles and shocks in cryptocurrency markets. International Review of Financial Analysis. 47, pp. 343-352 (2016)
4. Krafft, P., Penna, N.D., Pentland, A.: An Experimental Study of Cryptocurrency Market Dynamics. arXiv preprint arXiv:1801.05831 (2018)
5. Faste, T., & Faste, H.: Demystifying "design research": Design is not research, research is design. In: IDSA Education Symposium, vol. 2012, p. 15, (2012)
6. Walker, B. N., Nees, M. A.: Theory of Sonification. In Hermann, T., Hunt, A., Neuhoff, J. G., editors, The Sonification Handbook, pp. 9–39. Logos Publishing House, Berlin, Germany (2011)

7. Horn, R. E.: Information design: Emergence of a new profession. Information design, pp. 15-33 (1999)
8. Barrass, S., Worrall, D.: Welcome to ICAD from the Conference Co-Chairs, In: Proceedings of the 22nd Annual International Conference on Auditory Display, p. 3 (2016)
9. Hogg, B., Vickers, P.: Sonification abstraite/sonification concrete: An'aesthetic persepctive space'for classifying auditory displays in the ars musica domain. Georgia Institute of Technology. In: Proceedings of the 2006 Annual International Conference on Auditory Display (2006).
10. Hermann, T.: Taxonomy and definitions for sonification and auditory display. In Proceedings of the 14th International Conference on Auditory Display (2008).
11. Scaletti, C.: Sonification ≠ Music. The Oxford Handbook of Algorithmic Music, p. 363 Oxford (2017)
12. Supper, A.: Lobbying for the ear: The public fascination with and academic legitimacy of the sonification of scientific data (unpublished PhD thesis). Maastricht: Maastricht University 2012.
13. Roddy, S., Bridges, B.: Sound, Ecological Affordances and Embodied Mappings in Auditory Display In M. Filimowicz & V. Tzankova Editors (Eds.), New Directions in Third Wave Human-Computer Interaction: Volume 2 - Methodologies (pp.). New York, NY: Springer International Publishing 2018.
14. Bannon, L. J.: From Human Factors to Human Actors: The Role of Psychology and Human-Computer Interaction Studies in System Design. In: J. a. M. K. Greenbaum (Eds.),
Design at Work: Cooperative Design of Computer Systems, pp. 25-44 Lawrence Erlbaum, Hillsdale 1991
15. Bødker, S.: Third-wave HCI, 10 years later---participation and sharing interactions. 22(5), pp. 24-31. DOI: https://doi.org/10.1145/2804405
16. Roddy, S., Furlong, D.: Embodied aesthetics in auditory display. Organised Sound. 19(1), 70--77 (2014)
17. Barrass, S.: The aesthetic turn in sonification towards a social and cultural medium. AI & society, 27(2), 177--181, (2012)
18. Maurer IV, J. A.: The History of Algorithmic Composition. Center for Computer Research in Music and Acoustics. Stanford (1999)
19. Edwards, M.: Algorithmic composition: computational thinking in music. Communications of the ACM, 54(7), 58--67, (2011)
20. Eno, B.: Generative Music "Evolving metaphors, in my opinion, is what artists do." http://www.inmotionmagazine.com/eno1.html
21. Collins, N.: The analysis of generative music programs. Organised Sound, 13(3), 237-248 (2008)
22. Generative Music: http://www.generativemusic.com/
23. Toop, D.: The generation game: experimental music and digital culture. The Wire, 239-250, (2001)
24. Eno, B.: Discrete Music,(liner notes). Editions Eg Records, (1975)
25. Eno, B., Ziporyn, E., Gordon, M., Lang, D., & Wolfe, J.: Music for airports. Editions EG, (1978)
26. Scherzinger, M.: Curious Intersections, Uncommon Magic: Steve Reich's It's Gonna Rain. Current Musicology, (79/80), p. 207, (2005)
27. Pousman Z, Stasko J.: A taxonomy of ambient information systems: four patterns of design. InProceedings of the working conference on Advanced visual interfaces 2006 May 23, ACM, pp. 67-74, (2006)
28. Weiser, M. and Brown, J.S. Designing Calm Technology. PowerGrid Journal, 1:1, (1996)
29. Cox, C., Warner, D. eds. Audio Culture, Revised Edition: Readings in Modern Music. Bloomsbury Publishing USA, (2017).
30. Papadopoulos, G., Wiggins, G.: AI methods for algorithmic composition: A survey, a critical view and future prospects. In AISB Symposium on Musical Creativity. Vol. 124, pp. 110-117. Edinburgh, UK, (1999)
31. Cope, D.: Computer modeling of musical intelligence in EMI. Computer Music Journal, 16(2), 69-83, (1992)
32. Cope, D.: Virtual Music: Computer Synthesis of Musical Style Cambridge, Mass.: MIT Press, (2001)
33. Cope, D.: Computer Models of Musical Creativity. Cambridge, Mass.: MIT Press, (2006)

34. Briot, J. P., Hadjeres, G., Pachet, F.: Deep Learning Techniques for Music Generation-A Survey. arXiv preprint arXiv:1709.01620 (2017)
35. Miranda, E. R.: Evolutionary computer music London: Springer, (2007)
36. Boden, M. A.: The creative mind: Myths and mechanisms. Psychology Press, (2004)
37. Boden, M. A.: Creativity and art: Three roads to surprise. Oxford University Press, (2010)
38. McCormack, J., Eldridge, A., Dorin, A., McIlwain, Generative algorithms for making music: emergence, evolution, and ecosystems. In: The Oxford Handbook of Computer Music. Oxford (2009)
39. Collins, N., Brown, A.R.: Generative music editorial. Contemporary Music Review 28(1), 1–4, (2009)
40. Dorin, A., McCabe, J., McCormack, J., Monro, G., Whitelaw, M.: A framework for understanding generative art. Digital Creativity 23, no. 3(4) 239—259, (2012)
41. McDermott, J., Sherry, D., O'Reilly, U.: Evolutionary and generative music informs music HCI—and vice versa. In: Music and human-computer interaction, pp. 223-240. Springer, London, (2013)
42. Eigenfeldt, A. The human fingerprint in machine generated music. In Procceedings of xCoAx, (2013)
43. Loughran, R., O'Neill, M.: Generative music evaluation: why do we limit to 'human'. In Proceedings of the first Conference on Computer Simulation of Musical Creativity (CSMC 2016), Huddersfield, UK, (2016)
44. Roddy, S., & Bridges, B.: Sounding Human with Data: The Role of Embodied Conceptual Metaphors and Aesthetics in Representing and Exploring Data Sets. In Music Technology Workshop, (2016)
45. Malavolta, I.: Beyond native apps: web technologies to the rescue!(keynote). In: Proceedings of the 1st International Workshop on Mobile Development, pp. 1--2. ACM (2016)
46. Vukolić, M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In International Workshop on Open Problems in Network Security pp. 112—125, Springer Cham, (2015)
47. Neyer, G.: The future of blockchain. Journal of Digital Banking, 2(1), 74—96 (2017)
48. Kienle, H. M.: It's about time to take JavaScript (more) seriously. IEEE software, 27(3), (2010)
49. Tilkov, S., Vinoski, S.: Node. js: Using JavaScript to build high-performance network programs (2010).
50. Cabello, R.: Three.js. https://threejs.org/
51. Mann, Y.: Interactive music with tone.js. In: Proceedings of the 1st annual Web Audio Conference, (2015)
52. W3C. Introduction to Web Components. https://www.w3.org/TR/components-intro/
53. Polymer Project. https://www.polymer-project.org
54. Gat, I., Succi, G.: A Survey of the API Economy. Cut. Consort, (2013)
55. The Programmable Web https://www.programmableweb.com/
56. GDAX. https://www.gdax.com/
57. Roddy, S., Bridges, B.: Addressing the Mapping Problem in Sonic Information Design through Embodied Image Schemata, Conceptual Metaphors and Conceptual Blending. In Submission.
58. Subprotocol. Genetic JS. https://github.com/subprotocol/genetic-js
59. JeneticS.js http://jenetics.io/
60. mariosky. evospace.js https://github.com/mariosky/evospace-js
61. Merelo, J.: nodeo.js https://github.com/JJ/nodeo
62. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Kudlur, M.:. TensorFlow: A System for Large-Scale Machine Learning. In: OSDI, vol. 16, pp. 265—283, (2016)
63. Tensorflow.js. https://js.tensorflow.org/
64. van den Oord, A, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
65. Engel, K., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., Norouzi, M.: Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. 2017.
66. Lou, Q. Music Generation Using Neural Networks.
67. Magenta. Magenta.js https://magenta.tensorflow.org/js
68. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. arXiv preprint arXiv:1803.05428, (2018)

69. Clark, C., Tindale, A.:. Flocking: a framework for declarative music-making on the web. In: Proceedings of the 2014 International Computer Music Conference, (2014)
70. McCormick, C., Piquemal, S.: “WebPd” Available online at github.com/sebpiq/WebPd.
71. Roberts, C., Kuchera-Morin, J.: Gibber: Live coding audio in the browser. In: Proceedings of the ICMC, (2012)
72. Smalley, D.: Spectromorphology: explaining sound-shapes. Organised sound, 2(2), 107-126 (1997)
73. Flowers, J. H. (2005). Thirteen years of reflection on auditory graphing: Promises, pitfalls, and potential new directions. Georgia Institute of Technology.
74. Carter, S., & Nielsen, M. (2017). Using Artificial Intelligence to Augment Human Intelligence. *Distill*, *2*(12), e9.
75. Few, S.: Information dashboard design: The effective visual communication of data. Sebastopol, CA: O’Reilly Media, (2006)
76. Kitchin, R.: The real-time city? Big data and smart urbanism. *GeoJournal*, *79*(1), 1-14, (2014)
77. McArdle, G., Kitchin, R.: Improving the veracity of open and real-time urban data. *Built Environment*, *42*(3), 457-473, (2016)
78. Batty, M.: A perspective on city dashboards. *Regional Studies, Regional Science*, *2*(1), 29-32, (2015)
79. Sharda, R., Delen, D., Turban, E.: *Business intelligence: a managerial perspective on analytics*. Prentice Hall Press, (2013)
80. Korczak, J., Dudycz, H., Dyczkowski, M.: Design of financial knowledge in dashboard for SME managers. In *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on* (pp. 1123-1130). IEEE, (2013)
81. Yigitbasioglu, O. M.,Velcu, O.: A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, *13*(1), 41-59, (2012)
82. Velcu-Laitinen, O., Yigitbasioglu, O. M.: The Use of Dashboards in Performance Management: Evidence from Sales Managers. *International Journal of Digital Accounting Research*, *12*, (2012)