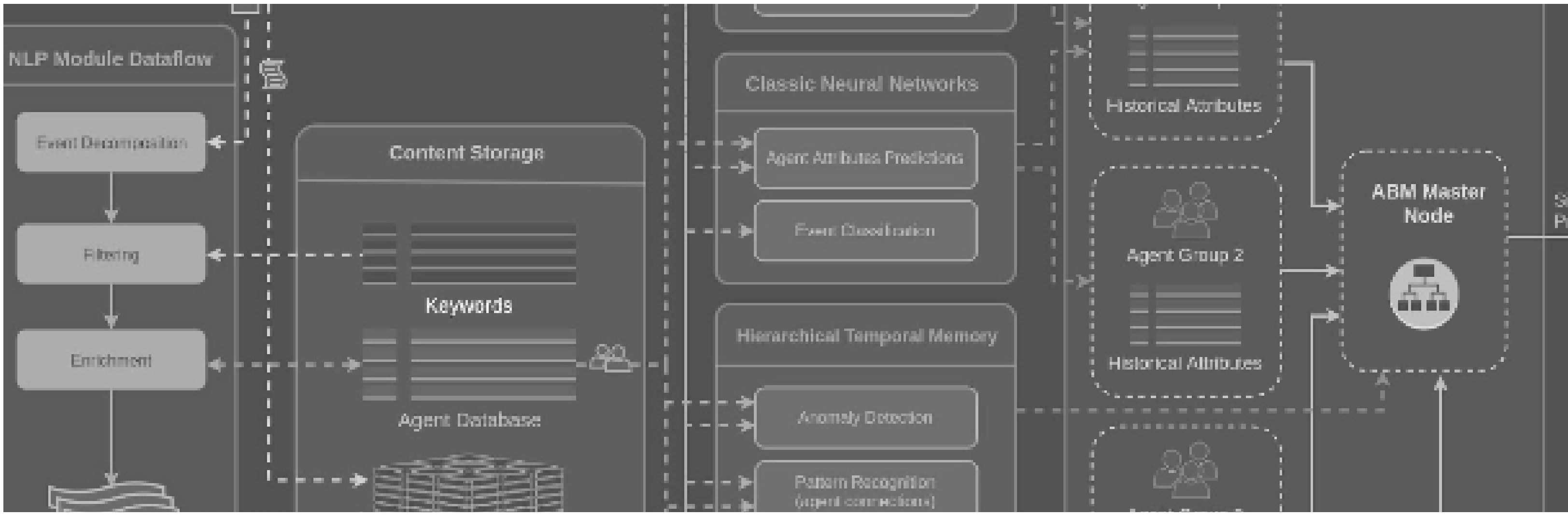
Yupana

Agent Based Modeling Framework for Cryptofinance



Abstract

The expansion of unstructured and uncorrelated data has outgrown the capabilities of conventional analytical models. Recent innovations in distributed ledger technology have led cryptocurrencies to become a rapidly growing asset class, now capturing the attention of investors, traders, developers, and regulatory bodies. These ecosystems emerge and proliferate incredibly fast, but require time, technical knowledge, analytical skills, and professional tools to properly understand. These properties make cryptocurrencies an ideal use case for the Yupana framework.

Yupana is a dynamic agent-based framework where large volumes of data from heterogeneous sources are mapped and go through machine and deep learning pipelines to extract useful patterns. We lay out a flexible framework that aims to become a community driven project, giving it the advantage of scalable development. By combining multiple methods of machine learning with natural language processing to inform an agent-based model, deeper insights can be extracted from the growing mass of unstructured information.

Global Vision

What stops us from making robust prediction systems in financial, and specifically crypto, markets? The game-changing nature of digital asset technologies requires identifying fundamentals instead of relying on preconceived notions that traditionally sit at the intersection of finance and contemporary machine learning. Generally speaking, there are three major concerns with any predictive modeling: computing power, data, and prediction models.

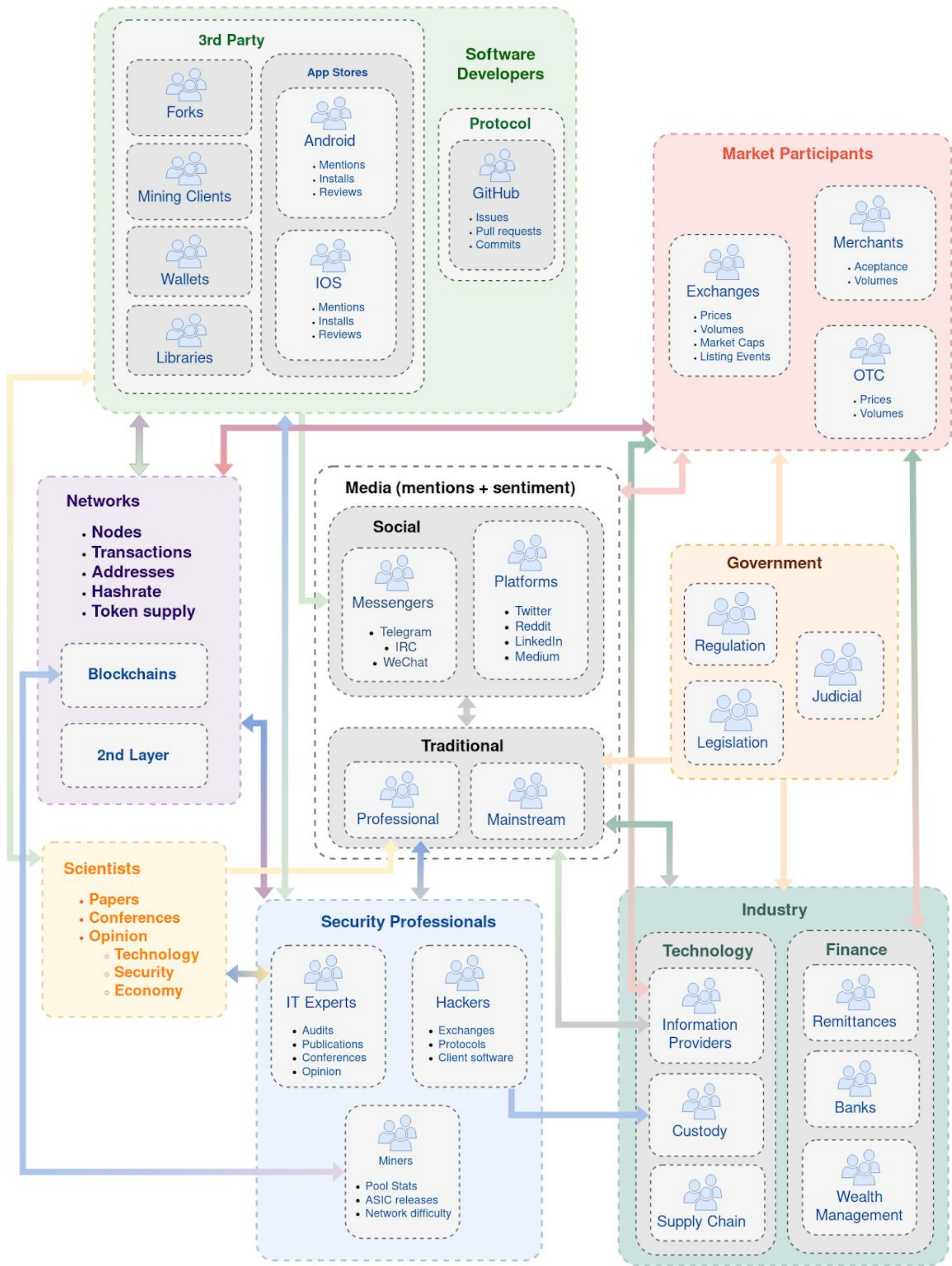
Computing power is relatively straightforward to address. The proliferation of cloud services has given users extreme amounts of computing power that can be easily scaled. Terabytes of financial, blockchain, and natural language data are readily available via native APIs as well as multiple data aggregators. The other two limitations have historically contributed to developmental setbacks of market prediction models. Instead of modeling financial markets as a singular interconnected information system, financial analysts use machine learning to solve specific use cases. Given the constant interventions by financial regulators and unreliable market instruments, often manipulated outside of free market rules, even the most sophisticated market simulation models break down over time.

Now - in the land where code is the law - crypto-native financial instruments provide a more solid foundation for machine analysis. The inviolable rules of cryptography and mathematics raise finance to the level of more established sciences like physics and biology, where AI technologies have decades of a head start. In addition to this, the open nature of digital asset projects and the surrounding ecosystem provide us with datasets that have been available to other sciences through simple observation. It is, therefore, our strong belief that the introduction of blockchain assets to financial markets requires a more fundamental analytical framework to take full advantage of the changing FinTech landscape.

ABM Approach

In conventional quantitative finance, analysts rely heavily on patterns derived from historical data using hand-picked features to produce trends in financial assets. While this approach can be successful in identifying general trends in representative periods of behavior, they are typically not powerful enough to provide an accurate signal on their own and must be heavily augmented with labor-intensive analysis. In addition to this, the unprecedented speed and scale at which new crypto-financial instruments are being developed makes any historical patterns risky to rely on: currently, even the most robust prediction models cannot fully adapt to the appearance of a new token, exchange, or financial regulation. A more holistic modeling approach is needed to address constant structural changes in cryptofinance.

A promising approach is modeling the whole ecosystem using a network of interconnected agents to produce simulations and/or predictions by aggregating and analysing the data they produce. This data can include, but is not limited to, common financial indicators such as market data, technical information from blockchain and project repositories, sentiment scores extracted from natural language data, etc.



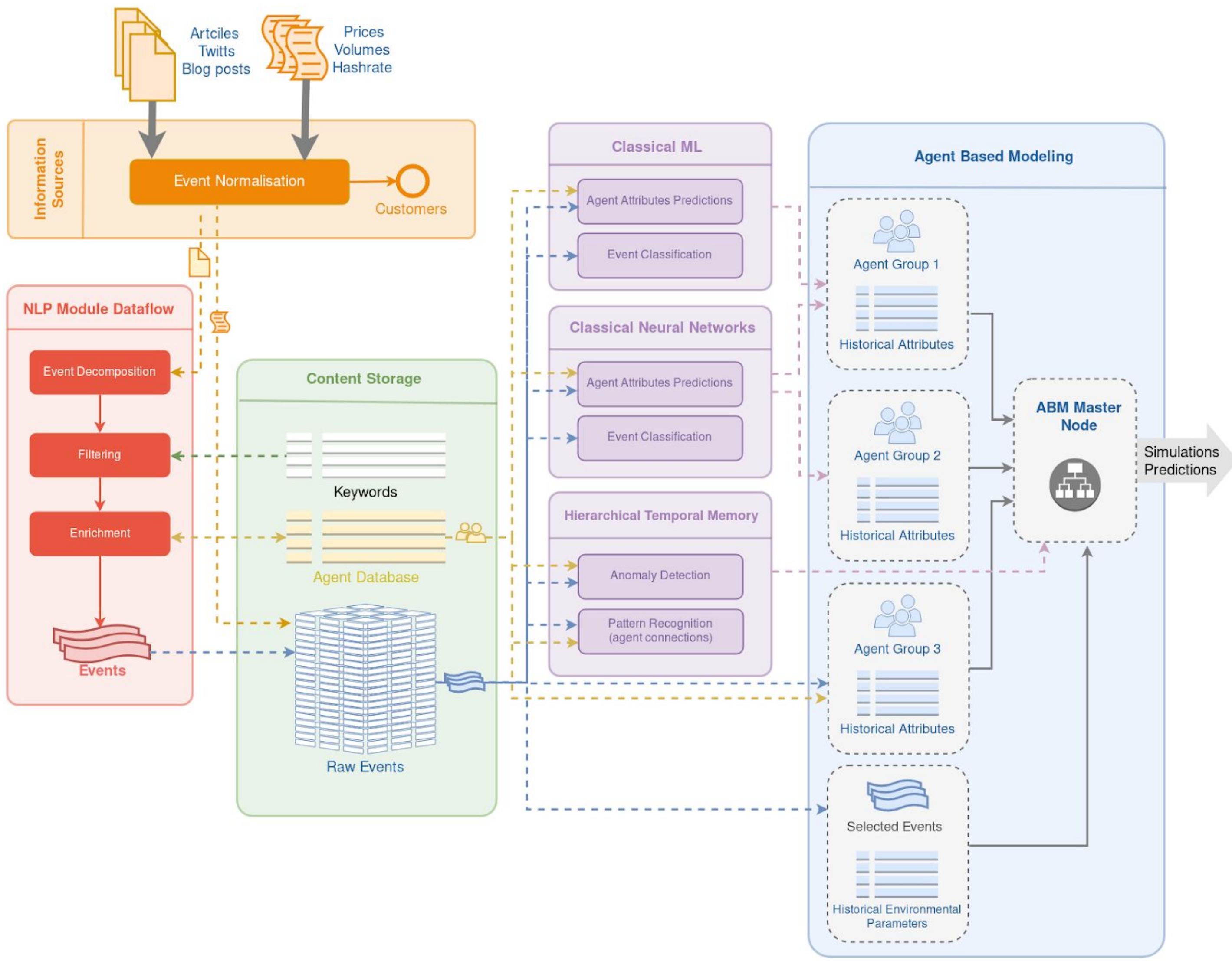
Agent Network Visualization

Interconnected agents are modeled as individual entities that represent actual market participants and their communities, exchanging information with each other through complex behavior. This behavior includes dynamic and flexible adoption to a changing scene, asynchronous and autonomous decisions, and efficient use of resources in a distributed environment. The agents are fed historical attributes data from upstream modules to outsource the necessary data transformations, filtering, normalizing, and applying of probabilistic models to calculate missing attributes. In addition to agent attributes, Yupana consumes network model information coming from modules that leverage graph theory and social network analysis to extract inter-relationships across different agents and agent communities.

It is worth noting that such an approach is not novel and is extensively being leveraged in areas such as evolutionary biology, logistics, and nuclear physics, where scientists and engineers are forced to abandon classic machine learning techniques to address the ecosystem complexity and the resulting computational limits. Biological ecosystem complexity approaches have been using interacting agents taking into account co-evolving traits per agent to understand spatial food webs and patterns of biological radiations. Studying seemingly trivial subjects such as whale migration requires modeling not only individual whale species, but also their interaction with a large range of organisms that surround and live inside of them. Being able to independently observe those organisms, record parameters, and map their behavior is key to making successful migration pattern predictions.

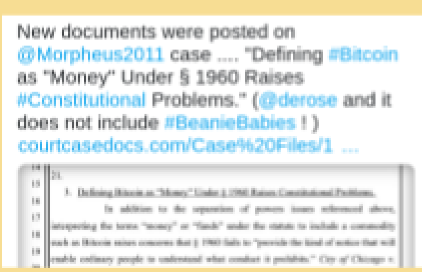
Yupana Architecture

The above described approach to ABM is already being adopted by Yupana. Yupana is a part of Nakamoto Terminal (***NTerminal***), a modular and flexible data aggregation and analysis platform. Its job is to provide unique insights in the digital currency space through real-time data aggregation and analysis for financial, blockchain, and natural language data. The computational potential and modular approach of Nakamoto Terminal enables analysis of large amounts of data under a complex multi-agent model. Its information pipeline is comprised of a set of modules around a message broker, which is essential for retrieving, converting, organizing, and delivering data. Each module contributes to the system by handling a specific task and delivering the processed results to the next component until the data is routed to Yupana, which applies agent based modeling to the pre-processed data.



Yupana Data Flow within NTerminal

Information Sources



Processing links, hashtags, mentions, and embedded images

The first step of the information pipeline is data gathering and normalisation, which is handled by the information source modules. They are named after and follow the logic of Source components within Java's Spring Framework, which NTerminal's information pipeline is based on. At this stage, relevant information gets spotted through different techniques, ranging from simple keyword filtering to blockchain address recognition. After the original content is recovered, it is normalised to adhere to information pipeline standards and passed along to information subscribers, which can include NTerminal's processors, sinks, or client-side latency-sensitive applications.

NLP Module

One of the major information subscribers is a group of natural language processors that handle social and traditional media, financial regulator announcements, messaging platforms, and other sources of human-generated information.

Decomposing documents and messages into annotated events

These processors work together to enrich extracted language data with quantifiable data points. By leveraging technologies, such as named entity extraction, optical character recognition, image annotation, and sentiment analysis, they turn documents, images, and audio information into meaningful data streams that can be later used to build prediction models and analyze market movements.

Content Storage

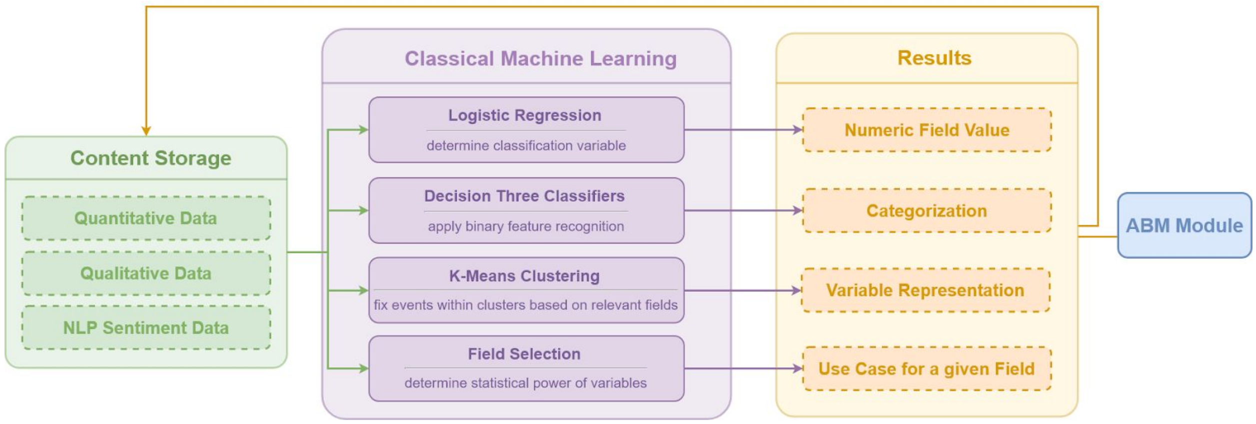
Enriching Agent Database with extracted named entities

While much of the information collected by NTerminal is available on the Splunk platform through REST APIs, as well as via flat file format, Yupana components also have access to 2 additional databases for information sharing. One is based on Elasticsearch and is used for storing processed text. This allows NTerminal modules to offload resource intensive keyword lookups and context extractions to the database itself. The other storage component is a classic relational database based on Postgres. NTerminal pipeline components mostly use it to store and share configuration settings, such as information resources to crawl, message routing, and running schedules. The NLP module leverages this configuration storage to collaborate on its keyword lists and Agent Database with other components.

Machine Learning Modules

Machine learning modules facilitate data transformations between content storage, agents, and network tools to synthesize input towards a functional form. Data transformations will work to simplify events in relation to previous data or outside agent activity, effectively contextualizing the data for simulating and learning pattern based network behavior.

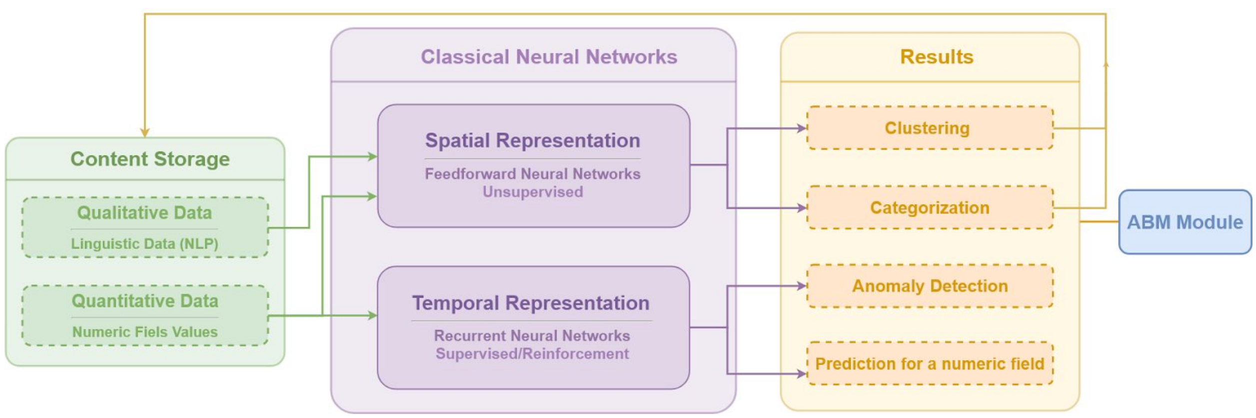
Classical ML



Simple algorithms, such as regression, decision trees, and clusters/classifiers are used for additional layers of preprocessing.

This layer is the initial processor, as it will be primarily founded on identification. This means that simple assumptions are made with regard to the relational and structural state of the data. This is a predominantly spatial representative layer that places data points in a relational perspective to their given fields and similar events.

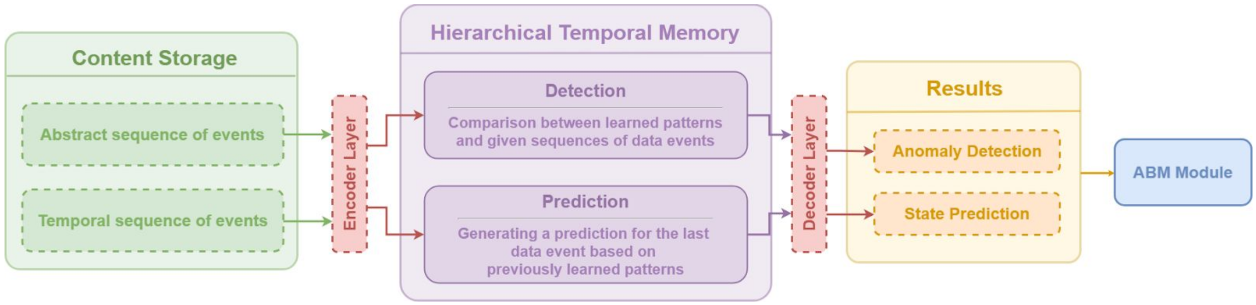
Classical Neural Networks



Classical implementation of neural network structures that will be trained on datasets for the development of temporal and spatial representations.

This layer will be focused on predicting a field or determining a value of a given variable based on a specific sensory input. A sensory input can be a specific data source such as Twitter sentiment over a given keyword. A temporal representation will take in a sequence of events that are structurally organized by the variable of time, and determine a future state of a given field, based on the previous inputs. This would be a time series analysis that could provide predictions or detect anomalies based on what is expected and what is observed. These neural networks can provide spatial classification or clustering of sequence data, given that the network has bidirectional attributes. This means that one can feed in a fixed sequence of data where each point is dependent on the previous node in the sequence, and therefore present a more robust classification.

HTM



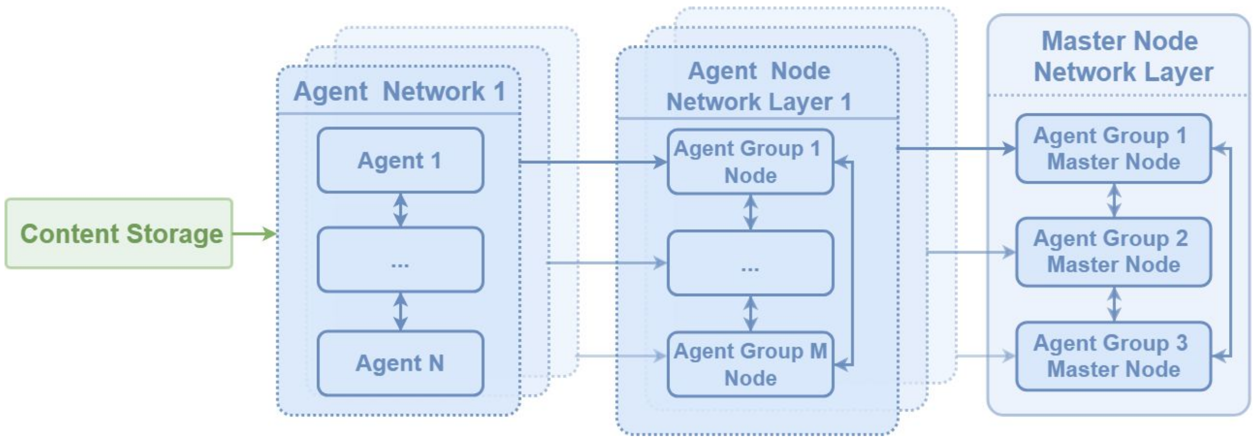
Abstract application of multisensory data relations along with predictive capabilities and anomaly detection.

This form of ML allows for sequential input to be structured into features that then form representative patterns in common activity, which allow for abstraction when a prediction is required at a given layer as well as detection of anomalous patterns occurring in the layering. This applies a stronger pattern mapping and predictive construction capability than classical Neural Networks. In addition, multiple data types can be brought together for multisensory processing of a given domain. This provides additional power to the prediction that is being made. The only caveat is that a given HTM implementation requires a significant repository of training data and needs to be structured in particular ways. However, given the power of it's predictions it will be an important tool for the aggregation of data to one specific claim for a given system.

ABM

High data volumes produced by individual agents will need to be classified, transformed, and put to network behaviors within the Yupana framework. Machine learning algorithms will act between data sources such as content storage, agents, and network behaviors, to convert event data into a usable format. Classical machine learning, neural networks, and hierarchical temporal memory techniques will be implemented to provide a diverse set of agent and network interactions.

The core of the ABM module is based on a general-purpose architecture that enables collaboration between different agents and the overlapping communities they belong to. The architecture accommodates heterogeneous agents that consume data from different sources by organizing them into multiple levels of groups. Agents with commonalities form functional clusters called *agent group nodes* which communicate to form a network layer. Utilizing reinforcement learning and computational voting techniques, these nodes can evaluate and provide feedback on the performance of the distributed agents within their cluster. Similarly, collections of agent group nodes, which characterize larger functions within the system, interact to form a higher-order "master node" network layer which provides feedback to their respective contributing nodes.



The ABM architecture is configured as three levels of connected networks: Agent Network Layers, Agent Node Network Layers, and an Agent Group Master Node Layer

Conclusion

We live in a financial world where, if it takes you more than a few moments to discover a 300-page document in Mandarin from a People's Bank of China website, translate it, understand its impact on the market, and deliver this actionable intelligence to your trading engine, you are already behind. The amount of data coming online is overwhelming current analytical tools that were built for the walled gardens of uniformly structured data.

To address this problem, Yupana employs various machine learning and data refinement techniques to inform its agent based modeling structure. By utilizing its network properties, it synthesizes otherwise uncorrelated information to derive meaningful insight. In this way, Yupana strives to provide a more comprehensive representation of a given system and scale to the needs of that system.

Yupana will initially be developed and refined by addressing the particularly well suited use case of crypto-financial analysis. That said, Yupana will later become an independent community-driven project. As data continues to open across industries, only through a decentralized and cross-disciplinary approach can Yupana scale successfully. The roadmap below lays out Yupana's development with this in mind:

NOW	NEXT	LATER
3-6 months	1-2 years	3-5 years
• Prototyping all components within Spring framework • Preparing historical data to train models • Testing the network using easily verifiable assumptions • Getting feedback on the concept from other industries and scientific community	• Fully integrating Yupana with Nakamoto Terminal Content Delivery Chain production environment • Open-sourcing all components and portions of the data • Presenting Yupana at scientific conferences • Expanding beyond cryptofinance, involving companies and industries	• Handing over project control to the open-source community • Spinning up Yupana modules as independent open projects

At the moment, Yupana's development is being supported by Inca Digital Securities' research division, IDS Labs, as well as a number of scientists and volunteers. We would like to extend an invitation to all parties interested in contributing to the development of this technology and using it to power real-world use cases. All research materials and prototypes can be found at https://gitlab.com/IncaSec/Labs/yupana/.