



GOLD

8 БОГИНО ХУГАЦААНЫ АРИЛЖААН ДАХЬ АЛТНЫ ҮНИЙГ ГҮН СУРГАЛТЫН АРГААР ТААМАГЛАХ ЗАГВАР

С.Дашзэвэг, М.Золжаргал

ХУРААНГУЙ:

Энэхүү судалгаанд гүн сургалтын арга ашиглан нэг унц алтны үнэ богино хугацаанд өсөх, буурах, тогтмол төлөвт байх гэсэн гурван ангиллаар таамаглал дэвшүүлж, LSTM загвартай харьцуулалт хийв. Алтны үнийн нийт 2.8 сая минутын тоон өгөгдлөөс онцлог шинж чанарыг ялган 1,689 өдрийн мэдээллийг загварын орц өгөгдлөөр түүвэрлэн авсан. Суурь загварт нууц давхарга дахь нейроны тоог бууруулахад сургалтын болон баталгаажуулах түүврийн алдааны утгуудын зөрүү 0.02–оор буурч, загварын үнэлгээ сайжирсан. Суурь загварт оролтын өгөгдлөөс 53–59 хувийн зөрүүтэй таамагласан. Суурь загварыг LSTM загвартай харьцуулахад илүү бага хувиар төөрсөн таамаглал дэвшүүлсэн. Энэхүү төрлийн загварыг хөгжүүлэхэд оролтын өгөгдлийн хэмжээг ихэсгэх, бусад эдийн засгийн ач холбогдолтой өндөр давтамжтай өгөгдлөөр өргөтгөх шаардлагатай байна.

Цаашид үнийн таамаглал дэвшүүлэх загварыг боловсронгуй болгосноор санхүүгийн эрсдэл, ашигт ажиллагаа зэрэг өргөн хэрэглээнд нэвтрүүлэх боломжтой юм.

Түлхүүр үгс: Гүн сургалт, Deep neural network, Үнийн таамаглал

JEL ангилал: E37, C52, C69

¹ СЗХ, ХШЗГ, Ахлах мэргэжилтэн, dashzeveg@frc.mn

² МУИС, МТЭС, Мэдээлэл компьютерийн ухааны тэнхим, Дэд профессор, zoljargal@num.edu.mn

АГУУЛГА

I.	УДИРТГАЛ	238
II.	СУДЛАГДСАН БАЙДАЛ БА ТААМАГЛАЛ	240
2.1	Олон улсын судалгаа ба үр дүн	240
2.2	Дэвшүүлж буй судалгааны таамаглал	242
III.	СУДАЛГААНЫ АРГА ЗҮЙ	242
3.1	Гүн сургалт	242
3.2	Олон давхаргат персептрон нейрон сүлжээ (Multilayer perceptron)	244
3.2.1	Идэвхижүүлэх функц (activation function)	244
3.2.2	Гаралтын функц	245
3.2.3	Хайперпараметр	245
3.3	Recurrent neural network	246
3.3.1	LSTM бүтэц	247
3.3.2	GRU бүтэц	248
3.4	Үнийн хөдөлгөөн таамаглах	248
3.5	Загварын өгөгдөл боловсруулах аргачлал	249
3.5.1	Үнийн хөдөлгөөн ба загварын гарц өгөгдөл сонгох	249
3.5.2	Техник шинжилгээний индикатор ба загварын орц өгөгдөл	250
3.6	Загварын алдаа үнэлэх арга	252
IV.	ӨГӨГДӨЛ БА ШИНЖИЛГЭЭ	252
4.1	Өгөгдөл	252
4.2	Дискретив статистик шинжилгээ	253
4.3	Онцлог шинж ялгах нь	255
4.4	Өгөгдөл боловсруулалт	257
V.	ЗАГВАР БА ТУРШИЛТ	258
5.1	Орчны тохиргоо	258
5.2	Суурь загвар (Baseline model)	258
5.3	LSTM загварын туршилт	260

ДҮГНЭЛТ	238
АШИГЛАСАН МАТЕРИАЛ	240
ХАВСРАЛТ	265
ХҮСНЭГТ, ЗУРГИЙН ЖАГСААЛТ	
Хүснэгт 1. Түүвэр өгөгдөл	252
Хүснэгт 2. Суурь загварын үнэлгээний утгууд	259
Хүснэгт 3. LSTM загварын үр дүн	260
Хүснэгт 4. LSTM үр дүнг бусад судалгааны үр дүнтэй харьцуулсан хүснэгт	261
Хүснэгт 5. Загварын харьцуулалт	262
Зураг 1. Нейроны торны энгийн бүтэц	243
Зураг 2. Нейроны торны математик загвар	243
Зураг 3. Нейроны идэвх	244
Зураг 4. RNN-ийн нэг зангилаа	247
Зураг 5. RNN-ийн зангилааг дэлгэвэл	247
Зураг 6. LSTM зангилааны бүтэц	247
Зураг 7. GRU зангилаа	248
Зураг 8. Арилжааны мөчлөг	249
Зураг 9. Алтны үнэ	251
Зураг 10. Алтны үнэ ба Бодлогын хүү	254
Зураг 11. Ерөнхий дискретив статистик, график	254
Зураг 12. Оролтын өгөгдлийн корреляцийн матриц	257
Зураг 13. DNN загваруудын алдагдлын утга ба ассурасу	259
Зураг 14. (256,128) загварын Ассурасу	261
Зураг 15. (128,64) загварын Ассурасу	261
Зураг 16. (64,32) загварын Ассурасу	261
Зураг 17. (256,128) таамагласан утга	261
Зураг 18. (128,64) таамагласан утга	261
Зураг 19. (64,32) таамагласан утга	261

I. УДИРТГАЛ

Үнэт цаасны зах зээлийн оролцогчдыг хөрөнгө оруулагчид (investor) болон арилжаачид (trader) гэж хоёр ангилж болно. Хөрөнгө оруулагчид нь 1 жил, 3 жил, 10 жил гэх мэт хугацаанд өгөөж өгөх хувьцаа, бонд, үүсмэл санхүүгийн хэрэгслийн арилжаанд оролцдог бол арилжаачид нь 1 цаг, 1 өдөр, 1 долоо хоног гэх мэт хэт маш богино хугацаанд үнийн өөрчлөлтөөс ашиг олох зорилгоор авах, зарах байдлаар арилжаа хийдэг.

Интернэт, ухаалаг төхөөрөмжүүдийн хэрэглээ нэмэгдэх тусам хөрөнгийн бирж рүү захиалга өгөх үе шатыг хялбарчилсан цахим арилжааны платформууд бий болж арилжаачдын тоо, арилжааны хэмжээ, дүн сүүлийн 10 жилд огцом нэмэгдсэн. Тухайлбал, “Монголын хөрөнгийн бирж” ТӨХК дээр явагдаж буй хувьцааны арилжаа 2013–2018 онд жилд дунджаар 61 тэрбум төгрөг байсан бол 2019–2023 онд жилд дунджаар 222 тэрбум төгрөг болж 3.6 дахин өссөн байна.

Өдрийн арилжаачид(day traders) –ын хувьд тухайн санхүүгийн хэрэгслийн үнийн талаар таамаглал дэвшүүлэхдээ суурь шинжилгээ буюу эдийн засгийн нөхцөл байдлын талаар үнэлгээ хийхээс илүүтэй техник шинжилгээнд тулгуурладаг. Техник шинжилгээ нь өнгөрсөн үеийн түүхэн үнийн статистик болон үнийн графикийн хэлбэрүүд (chart pattern) ашиглан зах зээлийн хандлагыг тодорхойлохыг зорьдог.

Үндэслэл

Үнийн графикийн хэлбэрийн шинжилгээ нь аливаа үнэ өсөх эсхүл буурахаасаа өмнө ямар нэгэн онцгой хэлбэр дүрс, эсхүл статистик шинж чанар үүсгэдэг гэсэн онол дээр тулгуурладаг. Тухайлбал, Доу онолд гурван удаа дээд ханшаа эвдсэн бол тренд эргэх нөхцөл бүрддэг гэх мэт байдаг. Тиймээс дохио, зураг зэрэг хэв танилтад (pattern recognition) ашигладаг гүн сургалтын арга техникийг үнийн таамаглал дэвшүүлэхэд ашиглах боломжтой байж болох юм.

Түүнчлэн өдрийн арилжаа хийдэг хүмүүст зориулан цагийн бүсийн хувьд түрүүлж явдаг Ази болон Европын арилжааны мөчлөгт үед гарсан үнийн хөдөлгөөний шинж чанарт тулгуурлан Нью-Йоркийн хөрөнгийн биржийн арилжааны 1 унц алтны үнэ өсөх эсэх таамаглалыг хийснээр зөв хөрөнгө оруулалт хийх суурь болно.

Зорилго, зорилт

Нью-Йоркийн хөрөнгийн биржийн арилжаа эхний 4 цагт 1(27гр) унц алтны үнэ өсөх, буурах, хэвэндээ байх гэсэн гурван төлөвийн аль нь байж болох тухай таамаглалыг Ази болон Европын арилжааны мөчлөгт гарсан хөдөлгөөнд тулгуурлан таамаглах боломжтой эсэхийг гүн сургалтын арга ашиглан тодорхойлно.

Судалгааны ажлын зорилтууд:

1. Нейроны тор, гүн сургалт, машин сургалтын арга техник ашиглан үнэ таамаглах чиглэлээр өмнө хийгдсэн

- судалгааны ажлууд судлах;
2. Арилжааны өгөгдөл цуглуулах, боловсруулах, шинж чанаруудыг ялган авч загвар үнэлэх алгоритм судлах;
 3. Өгөгдлийн шинжилгээ хийж чухал онцлогийг ялгах;
 4. Өгөгдлийг цэвэрлэж, сургалтын болон туршилтын өгөгдөл бэлтгэх;
 5. Загвар үүсгэх, үнэлэх, бодит арилжаанд туршихын тулд LSTM загвартай харьцуулах.

Судалгааны аргазүй

Эдийн засагчид, хөрөнгө оруулалтын шинжээчид үнийн таамаглал боловсруулахдаа эконометрикийн сонгодог загварууд (ARIMA, GARCH, VAR, SVAR) ашигладаг. Харин сүүлийн жилүүдэд илүү үр дүн үзүүлдэг ахисан түвшний алгоритмууд бий болж түүнийг дагалдах бэлэн сангуудыг ашиглахад хялбар болсон учир үнийн таамаглал боловсруулах цогц асуудлыг DNN, CNN, KNN, LSTM загварууд ашиглан шийдэхээр идэвхтэй оролдож байна.

Иймд өгөгдлийн шинжилгээ, тоон буюу эмпирик шинжилгээ, өгөгдлийн инженерчлэл, машин сургалтын загвар үүсгэх аргазүй, харьцуулан турших, ажиглалтын арга ашиглан судалгааны ажлыг гүйцэтгэнэ.

Ач холбогдол

Өдрийн арилжаа хийдэг хүмүүсийн хувьд Нью-Йоркийн арилжааны мөчлөгийн эхний 4 цагт ханш аль чиглэлд хөдлөх талаар тодорхой таамаглалтай байх нь арилжааны эрсдэлийн удирдлагыг сайжруулах маш чухал ач холбогдолтой.

Судалгаа талаас нь авч үзвэл сүүлийн 8 жилийн 1 минутаас илэрхийлсэн 2.88 сая их хэмжээний үнийн нарийвчилсан мэдээллээс чухал онцлогуудыг ямар байдлаар ялгаж авч болох талаар судалсан.

Судалгааны ажлын бүтэц

Энэхүү судалгааны ажлын судлагдсан байдал ба таамаглал хэсэгт өмнө үнэ таамаглал чиглэлээр хийгдэж байсан судалгааны ажил мөн дэвшүүлж буй судалгааны таамаглалын тухай, судалгааны арга зүй хэсэгт онол арга зүй, үнэ таамаглалын техник шинжилгээг хэрхэн уялдуулж болох тухай, өгөгдөл шинжилгээ хэсэгт тоон өгөгдлийг хэрхэн цуглуулсан, загварын оролтын өгөгдлийг хэрхэн бэлдсэн тухай, загвар ба туршилт хэсэгт орчны тохиргоо, суурь болон LSTM загварын үнэлгээний тухай, дүгнэлт загварын үр дүнг нэгтгэн дүгнэсэн тухай тус тус дурдана.

II. СУДЛАГДСАН БАЙДАЛ БА ТААМАГЛАЛ

2.1 ОЛОН УЛСЫН СУДАЛГАА БА ҮР ДҮН

Matloob(2020) нар машин сургалтын загвар ашиглан үнэ таамагласан судалгааны ажлуудын түүвэр цуглуулж нийт 512 судалгааны ажил хамруулж, үүнээс тухайн ажлын хэвлэгдсэн сэтгүүл, хэлэлцсэн хурал, тухайн судалгаанаас бусад судалгааны ажилд ишлэл авсан тоо зэрэг үзүүлэлтүүдээс хамааран 88 судалгааны ажлыг сонгон авч, ямар загвар ашигласан, үр дүн хэрхэн гарсан талаар харьцуулсан судалгаа хийсэн. Эдгээр 88 судалгааны ажлын 18 нь CNN, 38 нь LSTM, 9 нь DNN, 5 RNN, 8 нь Reinforcement learning аргуудыг ашигласан байх бөгөөд судалгааны ажлуудад таамаглалын үр дүнг хэмжихэд ашигласан RMSE, MAPE, MAE, MSE, accuracy, өгөөж гэх үзүүлэлтүүдийг харьцуулахад Reinforcement learning аргыг ашигласан судалгааны ажил илүү үр дүнтэй байсан. Мөн сүүлийн үед чиглүүлэгчтэй LSTM загварыг судалгааны ажлуудад илүү их хэрэглэсэн байна.

Zhong(2019) нар хебрид гүн сургалтын загварт S&P500 индексийн ханш болон эдийн засгийн бусад 60 үзүүлэлт ашиглан тус индексийн тухайн өдрийн хаалтын ханшийн таамаглал дэвшүүлсэн. Энгийн гүн сургалтын загварын нейроны тоог 10–аас 1000 болгон нэмэгдүүлэн үнэлэх туршилт хийхэд загварын нейроны тоо тодорхой цэгт хүрэхэд загварын алдаа буцаад өсөж, загварын таамаглах чадвар муудсан байна. Мөн загварын өгөгдөлд PCA (principal

component analysis) хувиргалт хийж загварыг үнэлэхэд хувиргалт хийгээгүй өгөгдөл ашигласан загвараас бага зэрэг сайжирсан байна.

Wataru(2019) нар Thompson reuters агентлагийн 2003–2013 оны мэдээллийн сангийн нийт 375 мянган нийтлэлийг гүн сургалтын загвар ашиглан эерэг, сөрөг утгатай мэдээлэл аль нь болохыг ялган авч, тухайн мэдээлэл нийтлэгдсэнээс хойш 1 цагийн дараа DJIA(Dow Jones Industrial Average) индекс хэрхэн өөрчлөгдөж болох тухай LSTM загвар ашиглан тооцоолсон. Тус судалгаанд сургалтын түүврийг хуваахдаа шаталсан хэлбэр(heirarchical)–ээр хуваах нь санамсаргүй хувааснаас илүү үр дүнтэй байсан байна.

Sai Krishna (2019) нар DJIA болон DJIA дээр нэмэлтээр газрын тос, алтны үнээс бүрдсэн хоёр тусдаа загварын өгөгдөл үүсгэн DJI индексийн тухайн өдрийн хаалтын ханшийн талаар SVM(support vector machine) болон LSTM(long short term memory) гүн сургалтын загварууд ашиглан таамаглал дэвшүүлсэн. SVM, LSTM загваруудыг тус тусад нь дээрх хоёр үндсэн өгөгдлийн хувьд үнэлсэн мөн үнийн хөдлөх дунжаар өргөтгөсөн өгөгдлөөр үнэлэхэд үндсэн өгөгдөл ашигласан LSTM загварын таамаглалын алдаа илүү сайн байсан байна. Тиймээс цаашид энэ төрлийн судалгааны ажлын чанарыг сайжруулахад

тоо шинж чанарыг ялган авах талаар зөвлөсөн байна.

Yoojeong(2018) нар Samsung Electronics Co., Ltd-ын хувьцааны үнийн нээлт, дээд, доод, хаалт, арилжсан хэмжээ зэрэг 5 үзүүлэлт ашиглан тус хувьцааны үнийн талаар 10 төрлийн үнийн хөдөлгөөний аль нь байж болох талаарх DNN загвар үнэлэхэд загварын accuracy 0.007 хувь байсан бол үнийн техник шинжилгээний индикаторууд (MA, MACD, VMA) –ийг ашиглан тус хувьцааны үнээс нийт 715 өөр төрлийн шинж чанар буюу индикатор тооцож үнэлгээ хийхэд accuracy 23.6 хувь болж сайжирсан бол 10 төрлийн үнийн хөдөлгөөний таамаглалаа 2 төрөл болгож өөрчлөхөд accuracy ойролцоо буюу 69.9 хувь гарсан байна. Уг судалгааны ажилд тухайлбал MA индикаторыг 5, 10, 20, 60, 120 өдрийн хөдлөх дунджаар тооцож, зах зээлийн хандлагын олон цагийн хамаарлыг загварт тусгах оролдлого хийсэн нь онцлог болсон байна.

Aurea Grane(2009) нар санхүүгийн зах зээл дээрх огцом хэлбэлзлийг түүврээс илрүүлэх асуудлыг хөрөнгийн зах зээлийн таамаглал дэвшүүлэхэд хөндөж түүврээс monte-carlo ашиглан өгөгдлийг цэвэрлэсэн ба GARCH загвар ашигласан.

Arman Khadjeh(2014) нар санхүүгийн зах зээлд нийтлэгдсэн мэдээний гол текстэд илүүц үг хэтэрхий их ордог учир зөвхөн гарчиг хэсгээс тухай мэдээлэл эерэг, сөрөг мэдээ болох тухай Хэл боловсруулалтын (Natural language processing) техник ашиглан тоон өгөгдөл болгон гүн сургалтын K-нп

болон SVM загварт оруулан валютын ханшийн хослолын талаар таамаглал дэвшүүлэхэд K-нп загварын accuracy 58.33 хувь гарсан бол SVM загварт 83.33 хувьтай үнэлэгдсэн.

Sidra(2021) судалгааны ажилд Энэтхэг улсын NIFTY 50 индекс буюу зах зээлийн үнэлгээгээрээ хамгийн том 50 компанийн индексийг ARIMA, LSTM, BiLSTM загвар ашиглан таамаглаж үзэхэд BiLSTM загварын үр дүн илүү сайн байсан тухай дурдсан байна.

Javen(2023) нар машин сургалтын арга ашиглан үнэ таамагласан судалгааны ажлуудын түүвэр судалгааг цуглуулан шинжлэхэдиймтөрлийнсудалгааныирээдүйн чиг хандлага NLP болон Reinforcement learning аргуудад илүү тулгуурлан хөгжихийг дурдсан байна. Тухайлбал загварын үнэлгээнд ашиглагдаагүй/ бүртгэгдээгүй тоо мэдээлэл буюу зах зээлд шинээр бий болсон өгөгдлийг тухайн загвар өндөр магадлалтай зөв таамаглал боловсруулж байвал сайн загвар боллоо гэж үздэг(model generalization). Сүүлийн үед хийгдсэн ихэнх судалгааны ажлууд нь судлаачдын тухайн түүврээс үүсгэсэн загварын гаралт өгөгдөлд тулгуурлан таамаглал боловсруулж байгаа тул загварын generalization тийм ч сайн биш байна. Тиймээс машин сургалтын загвар өөрөө загварын гаралт өгөгдлийн ангилал(classification) хийх чадамжтай загварыг боловсруулах нь илүү ач холбогдолтой болж байна.

2.2 ДЭВШҮҮЛЖ БУЙ СУДАЛГААНЫ ТААМАГЛАЛ

Өмнөх дэд бүлэгт дурдсан олон улсын судалгааны ажлуудын хамгийн сайн үр дүн (state-of-the-art) болон аргачлалуудыг харахад Reinforcement арга болон LSTM, BiLSTM загварууд хамгийн өндөр үр дүнтэй байна. Reinforcement аргыг бид хэрэгжүүлэхэд их хэмжээний өгөгдөл, тооцооллын өндөр хүчин чадалтай төхөөрөмж хэрэгтэй юм. Иймд бидэнд хангалттай өгөгдөл болон техникийн хүчин чадал байхгүй учир энэ удаагийн ажлаар уг аргыг дахин хэрэгжүүлэх боломжгүй байна.

Zhong(2019) –д DNN загварт өдрийн өгөгдлөөс шинж чанаруудыг ялган авч PCA хувиргалт хийж загварыг үнэлсэн бол энэхүү судалгааны ажилд минутын өндөр давтамжтай өгөгдлөөс шинж чанаруудыг

хоёр үе шаттайгаар ялган авч PCA хувиргалт хийн, үнийн таамаглал дэвшүүлнэ. Үүнийг суурь загвар гэж үзье.

Харин чиглүүлэгчтэй буюу LSTM, BiLSTM аргуудаар суурь загварт таамагласан үр дүнг шалгаж үзэж болох юм. Олон улсын судалгаанаас харахад эдгээр аргад өгөгдлийн инженерчлэл хийсэн нь бага байсан ба PCA зэрэг аргыг LSTM, BiLSTM-тай хамтатган ашиглаагүй байна. Иймд бид уг аргуудыг хамтатган үр дүнг шалгаж болно гэж таамаг дэвшүүлж байна.

Түүнчлэн практик ач холбогдлын хувьд цагийн зөрүүтэй бирж дээр арилжаалсан өгөгдөлд суурилан өөр биржийн үнийг тааж болно гэж үзэж байна.

III. СУДАЛГААНЫ АРГА ЗҮЙ

3.1 ГҮН СУРГАЛТ

Гүн сургалт (Deep Learning) бол тархины бүтцээс санаа авсан хиймэл нейроны тор (Artificial Neural Networks) –н дээр ажиллах алгоритмуудыг тухайлан судалдаг машин сургалтын нэг салбар юм.

Хүмүүс бид, маш их өгөгдөлтэй, тун хурдан компьютертой болсон нь гүн сургалт эрчимтэй хөгжих, гүн сургалтын загваруудыг сургах боломжийг нээсэн гэж зарим мэргэжилтнүүд үздэг. Хүмүүс туршлагаасаа

хэрхэн суралцдагтай адилаар гүн сургалтын алгоритмууд нэг үйлдлийг олон дахин давтаж хийхдээ, үр дүнгээ алхам бүрд бага багаар сайжруулж, суралцдаг. Энэ төрлийн алгоритмууд өгөгдөл ихсэхийн хэрээр улам сайн суралцаж, улам сайн ажилладаг.

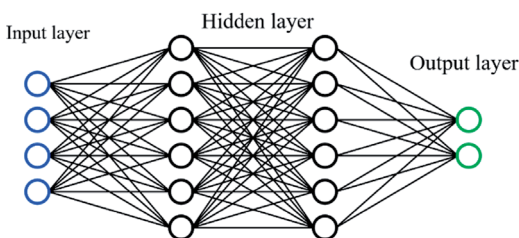
Гүн сургалтын чиглүүлэгчтэй төрөл (Supervised Learning) энэ салбарын гол амжилтыг бүтээж байгаа бөгөөд MLP, CNN, RNN зэрэг түгээмэл хэрэглэгддэг бүтэц,

түүнд суурилсан алгоритмуудтай байдаг. Гүн сургалтыг зураг боловсруулах (Image Processing), био-информатик, роботын удирдлага, санхүүгийн луйврыг илрүүлэхэд түгээмэл хэрэглээд зогсохгүй, цахим хэл боловсруулалт (NLP) –д ашиглах нь өндөр үр дүнтэй [19] байна.

Ямар ч даалгаварт зохицон ажиллах боломжтойгоос гадна гүн сургалтад байдаг өөр нэг давуу тал бол өгөгдлийн шинжийг автоматаар ялгаж авах чадвар юм. Үүнийг **Шинжийг ялгаж сурах** (Feature Learning) гэж нэрлэдэг. Өгөгдөлд анхнаасаа илэрхий өгөгдсөн доод түвшний шинжүүд дээр суурилан, өмнө нь тодорхойгүй байсан дээд түвшний шинжүүдийг таньж, энэ нь давхарга нэмэгдэх тусам улам төгөлдөр болдог. Олон олон давхарга дамжсан шинж чанарын энэ шаталсан бүтэц аливаа өгөгдлийг зөв ялгахад маш чухал үүрэгтэй.

Ийм нейроны тор, загвар бүтээх зорилго

Зураг 1. Нейроны торны энгийн бүтэц



$$h_{i+1} = \delta(W h_i + b) \quad (2.1)$$

Үүнд: δ – идэвхижүүлэх функц (activation function), w – нейроны жин(weights), b – гажилт(biases)

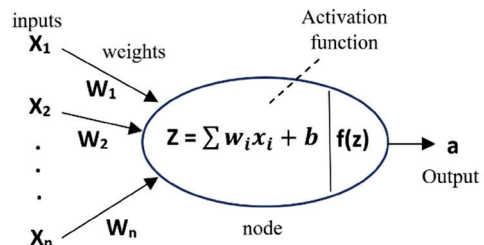
Хэрэв ганц нууц давхаргатай бол нэг давхаргат нейроны сүлжээ, хоёр ба түүнээс дээш нууц давхаргатай бол олон давхаргат

нь тархитай аль болох ижилхэн бүтэц байгуулахад биш, харин тооцоолоход хэцүү бодлогуудыг шийдэхэд хамгийн тохиромжтой алгоритм, нейроны торны болон өгөгдлийн бүтцийг туршиж олоход оршино.

Гүн сургалтыг бүрдүүлэгч үндсэн ухагдахуун нь нейроны тор юм. Нейроны тор нь хүний тархины бүтцээс санаа авч зохион бүтээсэн өөр хоорондоо холбоотой мэдрэлийн зангилаа (nodes) буюу нейронуудын нэгдэл юм (Зураг 1). Нейроны торонд оролтын давхарга (input layer), нууц давхарга (hidden layer), гаралтын давхарга (output layer) гэсэн үндсэн гурван давхарга байдаг.

Тус загвар нь h_i болон h_{i+1} хувьсагчдын хоорондох шугаман бус байдлыг нууц давхаргад байрлах нейронуудын тусламжтайгаар загварчилдаг. Нууц давхаргын нейрон бүр өөрийн гэсэн параметрууд (нейроны жин болон гажилт)–тэй байдаг [2].

Зураг 2. Нейроны торны математик загвар



нейроны сүлжээ буюу гүн нейрон сүлжээ/гүн сургалт (deep neural network/deep learning) гэж ангилдаг.

3.2 ОЛОН ДАВХАРГАТ ПЕРСЕПТРОН НЕЙРОН СҮЛЖЭЭ (Multilayer perceptron)

Нейроны торны тооцоолол гүйцэтгэх нэгж бол нейрон юм. Түүнийг “зангилаа”, “орой”, «нэгж», “нейрон” гэх мэтээр нэрлэдэг. Зангилаа нь бусад зангилаанаас болон гадаад эх үүсвэрээс оролтын мэдээлэл авч тооцоолол хийгээд, гаралт буцаана. Оролт бүрд жингийн утга (*weight* = *w*) дагалдах бөгөөд тухайн оролтын утга хэр чухал болохыг жингээр илэрхийлдэг.

Зангилаа нь, оролт бүрийн утгыг жингээр нь үржүүлээд хооронд нь нэмж, гарсан үр дүнг *f* функцээр өөрчлөөд гаралт руу өгдөг. Доорх зурагт хиймэл нейроны торны нэг зангилааны оролт ба гаралтын бүтцийг, хийх үйлдлийн хамт харуулсан байна.

Нейрон торны бодлогын нөхцөл:

Загварын орц өгөгдөл *u* болон таамаглахаар зорьж буй өгөгдөл τ -ын $\{u^n, \tau^n\}_{n=1}^N$ олонлогийн хувьд нейрон сүлжээний загварын функц $y=f(u)$ нь дараах байдлаар эрэмбэлэгдэнэ.

$$\begin{aligned} h_1 &= \delta_1(w_1u + b_1) \\ h_2 &= \delta_2(w_2h_1 + b_2) \\ &\vdots \\ y &= \delta_L(w_Lh_{L-1} + b_L) \end{aligned} \quad (2.2)$$

Энд: *L* – давхаргын тоо,

Алдааны функц ашиглан загварын нейроны параметруудийн оновчтой утгыг тодорхойлох:

Алдааны функц $\mathcal{E}(y^n, \tau^n)$ нь загварын анхны үнэлгээгээр тодорхойлогдсон утга болон таамаглахаар зорьж буй өгөгдөл τ^n -ын хоорондох зөрүүгээр тодорхойлогддог.

Загварын нийт параметр $\theta = \{w_1, \dots, w_L, b_1, \dots, b_L\}$ -ын утгуудыг

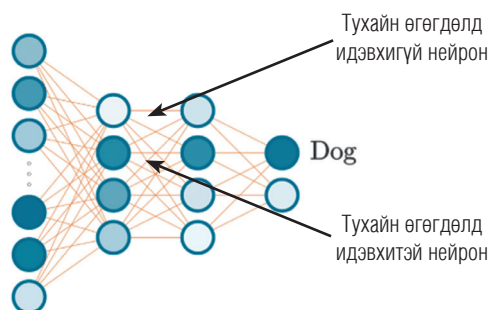
$$\min = \left[J = \sum_{n=1}^N \mathcal{E}(y^n, \tau^n) \right] \quad (2.3)$$

алдааны функц хамгийн бага байхаар оновчлол хийдэг. Функцийн оновчтой параметрыг тодорхойлохдоо градентын арга ашиглана.

3.2.1 Идэвхижүүлэх функц (activation function)

Гүн сургалтын үнэлгээнд хоёр төрлийн идэвхжүүлэх функц байдаг. 1) нууц давхаргын идэвхжүүлэх функц 2) гарц давхаргын идэвхжүүлэх функц. Энгийнээр тайлбарлавал тухайн нэг нейроны параметр буюу $output = weight * input + bias$ утга нь 1 бол дараагийн давхарга руу утгыг дамжуулна, 0 бол тухайн нейрон идэвхгүй буюу ажиллагаагүй болно.

Зураг 3. Нейроны идэвх



Нууц давхаргын идэвхжүүлэх функцүүд: Sigmoid эсхүл Relu(Rectified Linear Units)

функцуудын аль нэгийг түгээмэл ашигладаг.

Сигмойд идэвхижүүлэх функц:

$$y = \frac{1}{1+e^{-x}} \quad (2.4)$$

Relu идэвхижүүлэх функц:

$$y = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (2.5)$$

3.2.2 Гаралтын функц

Softmax функц нь гаралт матрицад гарч ирсэн утгуудыг Euler-ын тоо(2.71)-ын зэрэгт дэвшүүлэн, нормчлох замаар загварын гаралтын утгын аль нэгийг үнэн гэдгийг илүү сайн тодорхойлж өгдөг. Бидний хэрэгжүүлж буй даалгаврын хувьд, байгуулсан (нейроны торны) загварын гаралтын давхарга нь өсөх, буурах, хэвийн ангиллыг төлөөлөх гурван зангилаатай байна. Ангиллын тоо хоёроос их учраас олон ангилалд тохирсон *Softmax* ангилагч функц ашиглана. Зангилаа бүр дээрх бодит тоон гаралтын утга нь харгалзах ангиллын магадлалыг илэрхийлнэ.

$$S_{i,j} = \frac{e^{z_{i,j}}}{\sum_{j=1}^L e^{z_{i,j}}} \quad (2.6)$$

3.2.3 Хайперпараметр

Гүн нейроны торны бүтэц болон суралцах чадварыг тодорхойлдог хувьсагчуудыг хайперпараметрууд (hyperparameters) гэнэ. Сургалтын үйл явц эхлэхээс өмнө эдгээр хувьсагчийг тохируулж өгдөг. Сургалтын дараа гүйцэтгэлийн үр дүнгээс нь хамаарч хангалттай бол зогсоож, хангалтгүй бол дээрх хувьсагчдын утгыг өөрчилж, сургалтыг дахин шинээр эхлүүлэх байдлаар машин сургалтыг гүйцэтгэдэг.

Торны бүтэцтэй холбоотой хайперпараметрууд

- **Нууц давхарга болон нэгж (unit) –ийн тоо:** Оролтын давхарга болон гаралтын давхарга хооронд орших давхаргыг нууц давхарга гэнэ. Нейроны торны бүтцэд, туршилтын үр дүн сайжрахаа болих хүртэл хэдэн ч нууц давхарга нэмж болно. Нууц давхарга бүрд орших нэгжийн тоог зохицуулж, оновчлолыг сайжруулж болно. Нууц давхарга дахь нэгжийн тоо цөөн бол нейроны торонд *Underfitting* буюу огт суралцахгүй байх нөхцөл үүсдэг.
- **Гээх үйлдэл (dropout):** Хэт нийлэмж (*Overfitting*) буюу сургалтын өгөгдлийг л санаж, шинэ мэдээллийг суралцахгүй байх нөхцөлийг бууруулахын тулд гээх үйлдэл хийдэг. Гээх үйлдлийн утга 20%–50% хооронд байхад тохиромжтой. Өөрөөр хэлбэл нийт зангилааны сурсан мэдээллийн төдий хувийг тооцохгүй орхино гэсэн үг. Нууц давхаргууд олон, нейроны торны бүтэц том байх тусмаа гээх үйлдэл хийх хэрэгтэй. Ингэснээр суралцах чадвар сайжирдаг.
- **Жингийн эхний утга:** Тухайн давхаргад ашигласан идэвхжүүлэх функцээс хамаарч жингийн эхний утгууд ялгаатай байвал зохино. Ихэвчлэн жигд тархалттай жингийн утгыг ашиглана.
- **Идэвхжүүлэх/Ангилалх функц:** Гүн сургалтын загварт шугаман бус хамаарлыг шингээж өгөхийн тулд

шугаман бус идэвхжүүлэх функцийг ашигладаг. Хоёр ангилалтай бодлогод ашиглаж байгаа бол гаралтын давхаргад *Sigmoid* функцийг ашиглана. Хоёроос дээш тооны ангилалтай бодлогод ашиглаж байгаа бол *Softmax* функцийг ашиглана.

Сургалтын алгоритмтай холбоотой хайперпараметрууд

- **Суралцах хурд:** Торны жингүүдийг хэр хурдтайгаар шинэчлэх вэ гэдгийг энэ хувьсагчаар шийддэг. Суралцах хурд бага бол сургалтын явц удаан байх ч жингүүд тэмүүлсэн (зөв) утгадаа хүрч чадна. Суралцах хурд өндөр бол сургалтын явц хурдан ч тэмүүлсэн утгадаа хүрэхгүй байж болно.
- **Моментум:** Өмнөх алхмын мэдээлэл дээр үндэслэн дараагийн алхмынхаа чиглэлийг тодорхойлоход энэ

хувьсагчийг ашиглана. Тоон утга нь 0.5–0.9 байхад тохиромжтой.

- **Эргэлт (Epoch) –н тоо:** Энэ нь сургалтын нийт өгөгдлийг нейроны тор хэдэн удаа харах вэ гэдэг тоо юм. Шалгаж буй оновчлолыг сайжрахаа болих хүртэл эргэлтийн тоог нэмэгдүүлбэл зохино.
- **Багцын хэмжээ (Batch size):** Нэг удаад нейроны торонд өгөгдөх сургалтын өгөгдлийн түүврийг энэ хэмжээсээр тодорхойлно. Хамгийн тохиромжтой хэмжээ бол 32. Үүнээс цааш 64, 128, 256 гэх мэтээр нэмэгдүүлж болно.

Гүн нейроны торны оролтын давхарга, нууц давхаргууд, гаралтын давхарга, давхаргууд болон зангилаануудын хоорондын холбоос, жингүүд, хайперпараметрийн утгуудыг нэгтгээд **гүн сургалтын загвар** гэж нэрлэдэг.

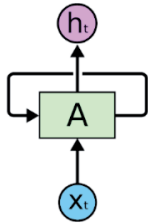
3.3 RECURRENT NEURAL NETWORK

Өгөгдлийн болон бодлогын шинж чанараас хамаарч гүн сургалтын аль хэлбэр, бүтцийг сонговол илүү үр дүнд хүрэх нь шийдэгддэг.

Цаг хугацаа болон орон зайн хувьд өмнө эсхүл хойно байрлах нь өгөгдлийн чанарт (утгад) нөлөөлдөг өгөгдлийг дараалсан (sequence) өгөгдөл гэнэ. Бичвэр бол дараалсан өгөгдлийн төрөлд хамаарна. Дараалсан өгөгдөлд одоогийн байдлаар хамгийн сайн ажилладаг гүн сургалтын загвар бол **Recurrent Neural Network (RNN)** нейроны тор юм.

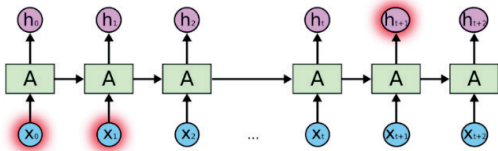
Доорх зурагт RNN нейроны торны нэг зангилааг харуулав. А нь x_t дэх оролтыг h_t гаралт руу дамжуулахдаа өмнөх алхмын ($t-1$) мэдээллийг 50%, энэ алхмын мэдээллийг 50% оруулж цааш дамжуулна. Дараагийн алхам ($t+1$) –д өмнөх хоёр алхмын мэдээлэл нийлээд 50% энэ алхмын мэдээллийг 50% оруулж дахин цааш дамжуулна. Дараагийн алхам ($t+2$)–д өмнөх гурван алхмын мэдээлэл нийлээд 50% энэ алхмын мэдээллийг 50% оруулж дахин цааш дамжуулна. Тор цаашид ийм байдлаар үргэлжилнэ.

Зураг 4. RNN-ийн нэг зангилаа



Хэрэв RNN торны энэ гинжин дарааллыг хангалттай урт хугацаанд үргэлжлүүлбэл эхэн үеийн мэдээлэл хойшлох тусам багасаж, бүр анзаарагдахгүй хэмжээнд хүртэл жижгэрнэ. Доорх зурагт үзүүлснээр x_0 болон x_1 оролтын мэдээлэл h_{t+1} гаралтад хүрэхгүй замхарч, харин x_{t+1} оролтын мэдээлэл өмнөх бүх алхмын оролтын мэдээлэлтэй тэнцэхүйц хэмжээтэй байна.

Зураг 5. RNN-ийн зангилааг дэлгэвэл



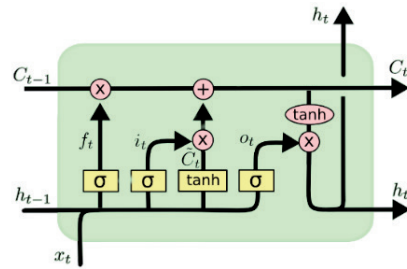
RNN торны энэ шинжийг *long-term dependency problem* гэх ба уг дутагдлыг засаж LSTM торыг Hochreiter нар 1997 онд бүтээсэн.

3.3.1 LSTM бүтэц

LSTM бүтцийн гол онцлог бол дараалсан зүй тогтлыг таньж, нейроны торны тухайн зангилаанд ирсэн хэрэгтэй мэдээллийг дараа дараагийн зангилаануудад дамжуулж, хэрэггүйг мартаж, эцсийн үр дүнд нөлөөлөх хүртэл хадгалах чадвар юм.

RNN нейроны торнууд бүгд дахин давтагдах модультай бөгөөд ердийн RNN зангилаа нэг \tanh давхаргатай байдаг бол LSTM зангилаа дөрвөн (3 *sigmoid*, 1 *tanh*) давхаргатай [18] байна. Доорх зурагт LSTM зангилааны үндсэн бүтцийг үзүүлэв. Зураг дахь зангилааны дээд хэсгээр хөндлөн чиглэлд нэвт гарсан ($C_{t-1}x + C_t$) шугам бол төлөвөө хадгалах санах ойг төлөөлнө. Үүн дээрх утгыг зангилааны төлөв (node state) гэнэ.

Зураг 6. LSTM зангилааны бүтэц



LSTM зангилаанд хийгдэх эхний үйлдэл бол h_{t-1} болон x_t оролтод ирсэн мэдээллээс хамаарч өмнөх алхмын мэдээллийг хадгалах уу, үгүй юу гэдгийг шийдэх юм. Үүнийг “мартах” хаалга (gate) гэнэ. σ (sigmoid) функцийн гаралт 0 ба 1-ийн аль нэг байх тул үр дүнд нь өмнөх алхмын мэдээлэл нэг бол “мартагдаж”, үгүй бол цааш дамжина.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.7)$$

Хоёр дахь үйлдэл бол одоогийн (t) алхмын аль мэдээллийг хадгалж үлдэх вэ гэдгийг шийдэх юм. Үүнийг “оролт” –ын хаалга гэнэ.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.8)$$

$$C'_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (2.9)$$

Гурав дахь үйлдэл бол зангилааны хуучин C_{t-1} мэдээллийг шинэ C_t мэдээллээр шинэчлэх.

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (2.10)$$

Хамгийн сүүлчийн үйлдэл бол гаралтад ямар мэдээлэл өгөх вэ гэдгийг шийдэх.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.11)$$

$$h_t = o_t * \tanh(C_t) \quad (2.12)$$

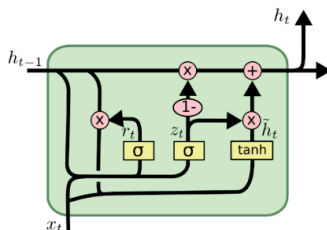
Энэ бол суурь бүтэц бөгөөд LSTM болгон яг ийм бүтэцтэй байх албагүй. Үндсэн үйлдэл болох хэрэгтэй мэдээллийг **сангах**, хэрэггүй мэдээллийг **мартах** чадварыг алдуулалгүйгээр уг бүтцэд дурын өөрчлөлт [18] хийж болно.

3.3.2 GRU бүтэц

Дотоод бүтцийн хувьд LSTM зангилаанаас нэлээд өөр боловч ажиллагааны үр дүнгийн хувьд адил, GRU бүтцийг 2014 онд Cho нар саналболгосон. Энэнь “мартах” болон “оролт” –ын хаалга (gate)–г нэгтгэж “шинэчлэх” хаалга болгосон бөгөөд зангилааны төлөв (node state) ба нууц төлөвийг h_t нэгтгэж,

мэдээллийн дотоод урсгалыг нь өөрчилсөн бүтэц юм. Зангилааны бүтэц нь LSTM–тэй харьцуулбал илүү энгийн болсон. Доорх зурагт GRU зангилааны бүтцийг харуулсан байна.

Зураг 7. GRU зангилаа



$$z_t = \sigma(W_z[h_{t-1}, x_t]) \quad (2.13)$$

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \quad (2.14)$$

$$h'_t = \tanh(W[h_tilde_t * h_{t-1}, x_t]) \quad (2.15)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h'_t \quad (2.16)$$

Нейроны тор дахь GRU зангилаа бүр дотроо дээрх үйлдлүүдийг (2.7, 2.8, 2.9, 2.10) дараалан гүйцэтгэж, тухайн оролт x_t болон, өмнөх нууц төлөв h_{t-1} –ийн утгаар одоогийн нууц төлөв h_t –ийг тооцон гаргана. Зангилааны бүтцээс харвал, оролт, гаралтын хэмжээс багассан нь нейроны торны нийт тооцоолох хурдад сайнаар нөлөөлнө.

3.4 ҮНИЙН ХӨДӨЛГӨӨН ТААМАГЛАХ

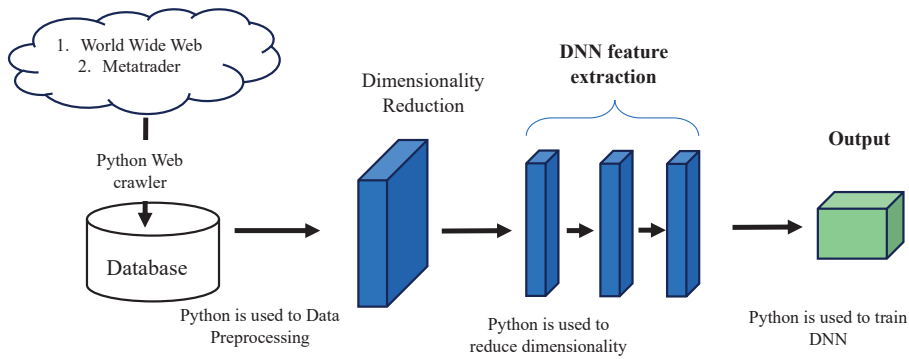
Өгөгдлийн онцлог шинж болох тухайн өдрийн арилжсан хэмжээ, үнийн дээд, доод утга, өмнөх өдөр, долоо хоног, саруудад үнэ ямар байсан гэх мэт шинж чанарыг агуулсан $[x_i, x_{i+1}, \dots, x_{i+m}]$ мэдээллийг ашиглан \hat{y}_{i+m+1} утгыг таамаглана. Загварын гарц утга \hat{y}_i –ын харьцуулах утга нь $[y_i, y_{i+1}, \dots, y_{i+m}]$ олонлогийн 0,1 гэсэн бинари матриц байна.

$$L = \sum_{i=1}^i \text{Loss}(\hat{y}_{i+m+1}, y_{i+m+1}) \quad (2.17)$$

Загварын үнэлэгдсэн утга болон харьцуулах утгын зөрүүг хамгийн бага байлгахаар загварын параметруудыг тооцоолно.

Загвар үнэлэхдээ дараах алхамыг хэрэгжүүлнэ. Үүнд:

1. Тоон өгөгдлийн ерөнхий статистик шинж чанаруудыг судална,
2. Загварт ашиглах өгөгдлийн шинж чанаруудыг ялган авна,
3. Өгөгдлийг загварын орц хэсэг болгох зорилгоор нормалчлана.
4. Өгөгдлийг сургалт (80%) болон тест (20%) гэсэн хоёр хэсэгт хуваана.
5. Загварын анхны хайпер параметруудийг сонгоно. Үүнд Dense layers, Activation function, Optimizer, Batchsize, learning rate, Epochs, dropout rate багтана.
6. Хамгийн их оновчтой таамагласан загварын параметруудийг ялган авч загвар үүсгэнэ.



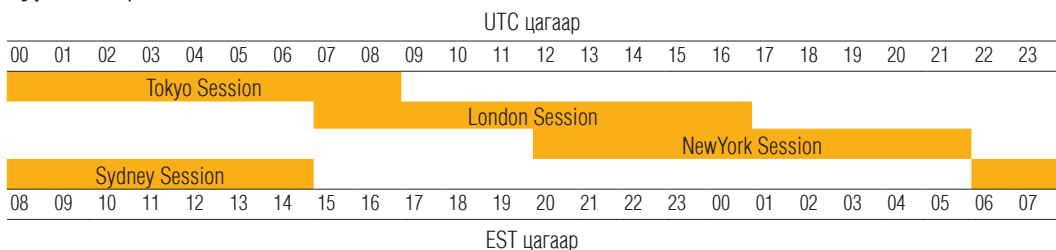
3.5 ЗАГВАРЫН ӨГӨГДӨЛ БОЛОВСРУУЛАХ АРГАЧЛАЛ

Энэхүү судалгааны ажлаар Нью-Йоркын арилжааны эхний 4 цагт гарах ханшийн хөдөлгөөнийг таамаглахаар зорьж байгаа. Арга зүй боловсруулахад хоёр төрлийн асуулт гарна: (1) ханшийн хөдөлгөөн гэж юуг хэлэх вэ? (2) хөдөлгөөнийг хэрхэн, ямар техник ашиглан таамаглах вэ?

3.5.1 Үнийн хөдөлгөөн ба загварын гарц өгөгдөл сонгох

Ази тивд хамгийн түрүүнд Даваа гаригийн өглөө Токио хотын хөрөнгийн бирж нээгдэж түүний дараа Хонгконг, Сингапур, Дели, Москва, Лондон, Нью-Йорк зэрэг томоохон хотуудын биржүүд цагийн бүсээс хамааран нээгдсэнээр ажлын 5 хоног, 24 цаг тасралтгүй арилжаа явагдаж байдаг.

Зураг 8. Арилжааны мөчлөг



Нью-Йоркийн арилжааны эхний 4 цагт гарах хөдөлгөөн гэдэг нь Нью-Йоркийн бирж нээгдэх ханш ба түүнээс 4 цагийн дараах ханшийн зөрүүгээр тодорхойлогдоно. Улаанбаатар хотын цагаар 20:00:00 болон шөнийн 00:00:00 цагийн хооронд үүссэн ханшийн зөрүү, арилжааны платформ (metatrader5)-ын 16:00:00 ба 20:00:00 цагийн зөрүү болж байгаа юм. (Wataru, Irena, 2019) үнийн хөдөлгөөнийг хоёр өөр цагийн хоорон дах ханшийн харьцаагаар шууд авах нь үр дүнтэй биш бөгөөд тоон ямар утгаас дээш хөдөлгөөнийг өссөн эсхүл буурсан гэж үзэх тухай дурдсан байдаг. Тиймээс энэхүү ажилд ханшийн хөдөлгөөнд тоон утгын хязгаар(price change threshold) тавьж өгнө[7].

Хоёр хугацааны үнийн харьцааны нийт хугацаан цувааны кватрил 80 буюу Q80%-аас дээш бол үнэ өссөн буюу эерэг хөдөлгөөн гарсан хэрэв кватрил 20 буюу Q20%-аас бага бол буурсан буюу сөрөг хөдөлгөөн гарсан. Харин Q20-Q80 хооронд бол ханш тогтмол буюу онц ач холбогдолтой хөдөлгөөн гараагүй гэж үзье.

$$y_i = \frac{P_{a,i}}{P_{b,i}} = \begin{cases} 1 \text{ or positive up if } s_i > Q80 \\ 2 \text{ or negative down if } s_i < Q20 \\ 0 \text{ or neural if otherwise} \end{cases}$$

3.5.2 Техник шинжилгээний индикатор ба загварын орц өгөгдөл

Үнэт цаасны арилжаа хийдэг хүмүүсийн хувьд зах зээлийн талаар таамаглал дэвшүүлэхдээ 1) суурь шинжилгээ, 2) техник шинжилгээ ашигладаг. Энэхүү судалгаанд үнийн талаарх техник шинжилгээний индикатор үзүүлэлтүүдийг загварын оролтын өгөгдлөөр ашиглана.

Гулсах дундаж (moving average): Гулсах дундаж гэдэг нь тодорхой хугацааны туршид үнийн өгөгдлийг жигнэж дундаж гаргах статистик арга юм. Энэ нь санхүүгийн зах зээл дээр хөрөнгийн үнийн хэлбэлзлийг жигд болгох, чиг хандлагыг тодорхойлох зорилгоор ашиглагддаг. Алтны минутын хугацаан цувааны үнийн гулсах дундаж 120 гэж тооцвол сүүлийн 2 цагт дунджаар ямар хандлагатай байсныг илэрхийлнэ. Энэхүү гулсах дунджийг зах зээлийн хандлага тодорхойлох зорилгоор ашиглана.

$$PMA = \frac{1}{n} \sum_{i=1}^n Closeprice_{t-n}^s ; \quad (2.19)$$

$$VMA = \frac{1}{n} \sum_{i=1}^n Volume_{t-n}^s \quad (2.20)$$

Энд: PMA – үнийн хөдлөх дундаж, VMA – арилжсан тоо хэмжээ, n – хөдлөх дундаж тооцох хугацааны хоцрогдол, s – тухайн үнэт цаасны төрөл

Дээрх үзүүлэлтүүдийг минут, өдөр, 7 хоног, сар гэсэн хугацааны хувьд тус бүрийн PMA , VMA (120, 240, 720) гэсэн хоцролтойгоор тооцно.

Хандлага тодорхойлох нь: t болон $t+1$ хугацааны хоорон дахь гулсах дунджийн өсөлтийн хувь нь эерэг байвал зах зээл өсөх хандлага давамгайлж байна, хэрэв сөрөг байвал зах зээлд унах хандлага давамгайлж байна гэж үзнэ (2.21), мөн уг үзүүлэлтийн тодорхой хугацааны нийлбэр утгаар тодорхойлж болно (2.22). Энэ нь урт хугацааны хандлага өөрчлөгдөж байгаа эсэхийг тодорхойлоход чухал мэдээлэл болно.

$$Gradk_t^s = \frac{MAk_t^s - MAk_{t-1}^s}{MAk_{t-1}^s} ; \quad (2.21)$$

$$SumGk_t^s = \sum_{k=0}^{39} Gradk_t^s (k = 60, 120) \quad (2.22)$$

Арилжааны идэвх тодорхойлох нь:

Арилжаалсан хэмжээний гулсах дунджийн өсөлт буюу VMA нь эерэг байвал зах зээлд арилжааны идэвх нэмэгдэж байна гэж үзнэ. Өөрөөр хэлбэл зах зээлийн идэвх тодорхой хугацааны интервалд явагдаж байсан хэмжээнээс ямар нэгэн мэдээ эсхүл зах зээлийн өөрчлөлтөөс хамааран идэвх суларч эсхүл сайжирсан байгааг илтгэнэ. Тухайлбал Америкийн Төв банк бодлогын хүүгийн шийдвэр гаргахаас 2–3 хоногийн өмнө бодлогын хүүгийн шийдвэрийг ямар гарах бол гэж зах зээл хүлээнэсээс үүдэн арилжааны идэвх эрс сулрах гэх мэт арилжааны идэвхэд нөлөөлдөг.

$$VGradk_t^s = \frac{VMA_t^s - VMA_{t-1}^s}{VMA_{t-1}^s} \quad (2.23)$$

Алтан огтлолцол (Golden cross):

Зах зээлийн хандлагын эргэлтийн цэгийг урт болон богино хугацааны гулсах дунджууд

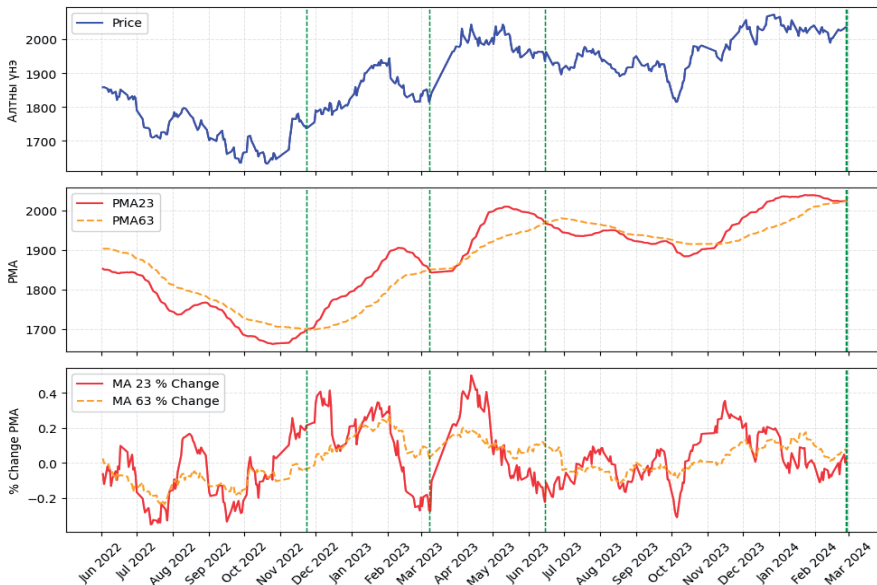
огтлолцсон эсэхээр тодорхойлж болдог. $t-1$ хугацаанд MA5 буюу богино хугацааны гулсах дунджийн утга нь MA10 буюу арай илүү урт хугацааны гулсах дунджаас бага байсан гэж үзвэл богино хугацаанд зах зээл уналттай байсан гэж үздэг. Харин t хугацаанд MA5 нь MA10-ын утгаас их болсон бол богино хугацаанд өсөлттэй болж байна буюу зах зээлийн унах хандлага эргэж өсөх хандлага руу шилжиж байж болзошгүй гэж үздэг.

$$ArrGk = \begin{cases} True & \text{if } (MA5_{t-1}^s < MA10_{t-1}^{10}) \\ False & \text{otherwise} \end{cases}$$

and $(MA5_t^s < MA10_t^{10})$

Дээрх арга томьёонуудыг жишээ болгож 2022.06.01–ээс 2024.03.29 хүртэлх өдрийн алтны үнэ ба гулсах дундаж 23, 63 өдөр, PMA23, PMA63-ын өсөлтийн хувиар илэрхийлсэн харьцуулах график байгуулж Зураг 9–д үзүүлэв.

Зураг 9. Алтны үнэ



3.6 ЗАГВАРЫН АЛДАА ҮНЭЛЭХ АРГА

Загварын таамаглах чадварыг бусад загвартай харьцуулахдаа таамагласан утга болон бодит утгын зөрүү буюу бодит байдлаас хэр зэрэг зөрүүтэй таамаглаж байгаагаар тооцдог. Алдааны утгыг дараах байдлаар тооцож болно. Үүнд:

MSE (Mean Squared Error)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.25)$$

RMSE (Root Mean Squared Error)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.26)$$

MAE (Mean Absolute Error)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.27)$$

MAPE (Mean Absolute Percentage Error)

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.28)$$

IV. ӨГӨГДӨЛ БА ШИНЖИЛГЭЭ

4.1 ӨГӨГДӨЛ

Судалгааны ажлын гол тоон өгөгдлийн түүврээр Metatrader5 арилжааны флатформоос 2016.01.11-ны өдрийн 01:00:00 цаг, минутаас 2024.02.29-ны өдрийн 11:41:00 цаг хүртэл нийт 2,882,154 минутын нэг унц алтны ам.доллар илэрхийлсэн үнийг авч ашигласан. Туслах өгөгдлөөр санхүүгийн зах зээлийн хандлагыг тодорхойлоход чухал

нөлөө үзүүлдэг АНУ-ын Холбооны нөөцийн сангаас зарладаг мөнгөний бодлогын (FED) хүү, мөн тус улсын Статистикийн хорооноос зарладаг хэрэглээний үнийн индекс (ХҮИ)-ийг сонгон авсан. Дээрх түүвэрт өгөгдлийн шинжилгээ хийж загварт ашиглах боломжит тоон өгөгдлийг гаргаж авсан.

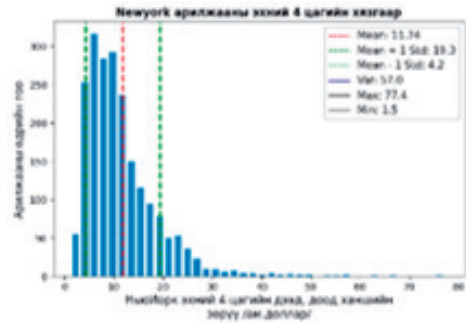
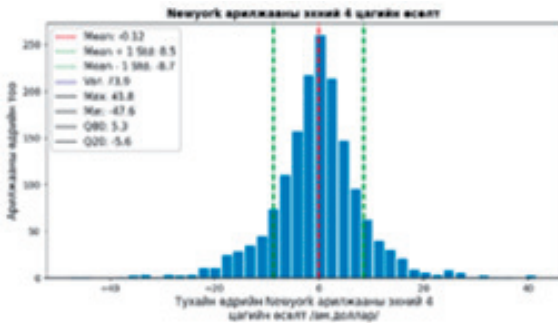
Хүснэгт 1. Түүвэр өгөгдөл

Д/д	Огноо	Нээлт	Дээд	Доод	Хаалт	Тухайн минутад үнэ өөрчлөгдсөн тоо	Арилжсан хэмжээ
1	2016-01-11 01:00:00	1103.82	1103.82	1104.27	1104.02	73	0
2	2016-01-11 01:01:00	1104.01	1105.92	1104.0	1105.92	78	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
2,882,152	2024-02-29 11:40:00	2035.03	2035.05	2034.92	2034.96	307	24560
2,882,153	2024-02-29 11:41:00	2034.96	2034.98	2034.83	2034.84	47	3760

Эх сурвалж: MetaTrader5 platform

Нью-Йоркийн арилжааны эхний 4 цагт Алтны дээд болон доод үнэ дунджаар 11 долларын зөрүүтэй, хамгийн ихдээ абсолют утгаараа 77 доллар хөдөлгөөн гарч байсан байна. Харин яг 4 цагийн дараах хөдөлгөөн

хамгийн ихдээ 41 доллар өссөн, хамгийн ихдээ 47 доллароор буурсан байна. Ашиглаж буй өгөгдөл нийт 1,689 өдрийн өгөгдөл орох бөгөөд үүнээс өссөн нийт 467, буурсан 468, хэвийн 753 өдөр тооцогдож байна.



Дээрх өгөгдөлд нийт 1,689 өдөр бүртгэгдсэн бөгөөд үүнээс хэвийн байх 753, өсөх 467, буурах 469 буюу тэнцвэртэй бус өгөгдөл үүсэж байгаа учир хэвийн байх 753 өгөгдлөөс

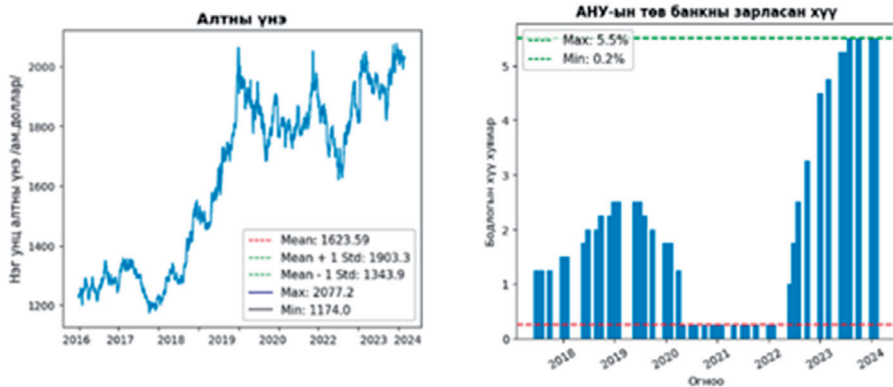
санамсаргүй байдлаар 470 өгөгдлийг сонгож ангиллыг 470, 469, 468 байхаар тэнцвэрт байдалд шилжүүлсэн.

4.2 ДИСКРЕВТИВ СТАТИСТИК ШИНЖИЛГЭЭ

Алтны үнэ 2023 оны эцэст өмнөх оны мөн үеэс 13.1 хувиар өсөж 2063 ам.долларт хүрсэн бөгөөд сүүлийн 4 жилд 1620–оос 2100 долларын хооронд арилжаалагдсан байна. АНУ–ын Төв банкны Мөнгөний бодлогын хорооноос 2022 оны 1 сарын 26–ны өдөр хэрэглээний үнийн өсөлтийг бууруулах зорилгоор бодлогын хүүг 0.25 хувиар өсгөж бодлогын хүүгээ 0.5 хувь болгох шийдвэр гаргасан бөгөөд энэ нь

сүүлийн 16 удаагийн хурлаар 0.25 хувьд тогтмол барьж байсан мөнгөний бодлогод эрс өөрчлөлт гарч, зах зээлд маш их хэлбэлзэл бий болгосон томоохон шийдвэр байсан. Түүнээс хойш нийт 16 удаагийн Мөнгөний бодлогын хорооны хурлаас 11 удаад нь бодлогын хүүг өсгөх шийдвэр гаргаж 2024 оны 1 сарын 31–ны байдлаар 5.5 хувьд хүрсэн байна.

Зураг 10. Алтны үнэ ба Бодлогын хүү

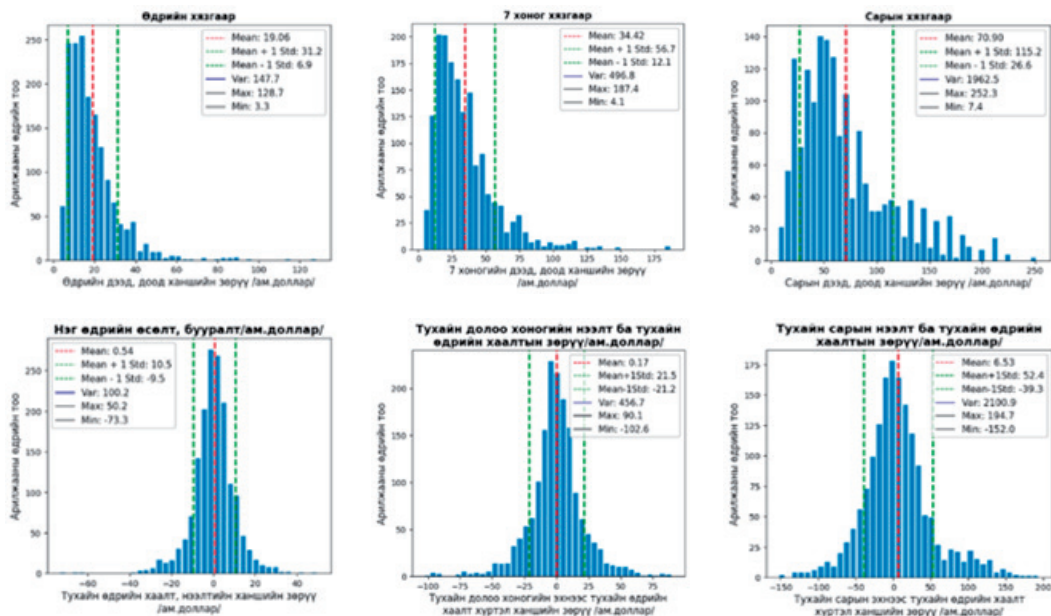


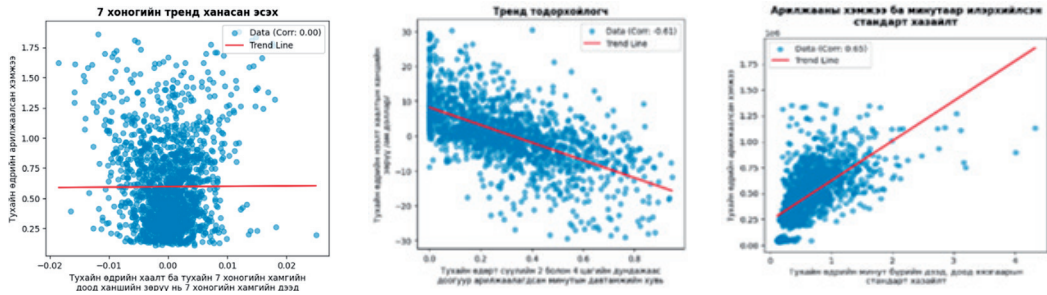
Эх сурвалж: MetaTrader5 platform

Харин АНУ-ын хэрэглээний үнийн жилийн өсөлт 2021 оны 4 сараас эхлэн 3 хувьд хүрч эхэлсэн бөгөөд 2021 оны төгсгөлд 5.42 хувьд хүрсэн байна. Энэ нь бодлогын хүүг өсгөх суурь нөхцөл болсон. Алтны үнэ сүүлийн 8

жилийн хугацаанд Азийн зах зээл дээр дээд, доод үнийн зөрүү нэг өдөрт хамгийн ихдээ 127 доллар, 7 хоногт 187 доллар, нэг сарын хугацаанд 252 доллар хүртэл хэлбэлзэл үзүүлсэн байна (Зураг 11).

Зураг 11. Ерөнхий дискретив статистик, график





Эх сурвалж: Судлаачийн тооцоолол

Харин хаалт болон нээлтийн үнийн зөрүү буюу өсөлтийн түвшин өдөрт хамгийн ихдээ 50 доллар өсөж, 73 доллароор унасан, 7 хоногийн хугацаанд хамгийн ихдээ 90 доллараар өсөж, 102 доллароор унасан харин нэг сарын хугацаанд хамгийн ихдээ 194 доллароор өсөж, 152 доллароор унаж байсан байна (Зураг 11). Дээрх графикаас алтны үнэ өдөрт дунджаар 19 доллар, 7 хоногт дунджаар 34 доллар, сард дунджаар 70 долларын хөдөлгөөн үзүүлдэг байна. Тренд тодорхойлогч буюу зураг 11–т тухайн өдөрт 2 болон 4 цагийн гулсах дунджаас үнэ тогтсон минутын тоо өндөр байх тусам хаалт,

нээлтийн зөрүү сөрөг утгатай болохыг илтгэж байна. Өөрөөр хэлбэл 2 болон 4 цагийн гулсах дунджаас доогуур хэт олон удаа арилжаалагдах нь хаалтын ханш нээлтийн ханшаас доогуур хаагдах магадлалыг өндөр болгохыг тодорхойлж байна. Зураг 11–т арилжаалсан хэмжээ нэмэгдэх болон богино хугацаан дах үнийн хөдөлгөөний стандарт хазайлт эерэг утгатай болохыг харуулж байна. Өөрөөр хэлбэл үнийн хэт хэлбэлзэл үүсэхэд зах зээлийн идэвх сайжирдаг байж болох эсхүл зах зээлийн идэвх нэмэгдэхэд үнийн хэт хэлбэлзэл бий болдог байж болох юм.

4.3 ОНЦЛОГ ШИНЖ ЯЛГАХ НЬ

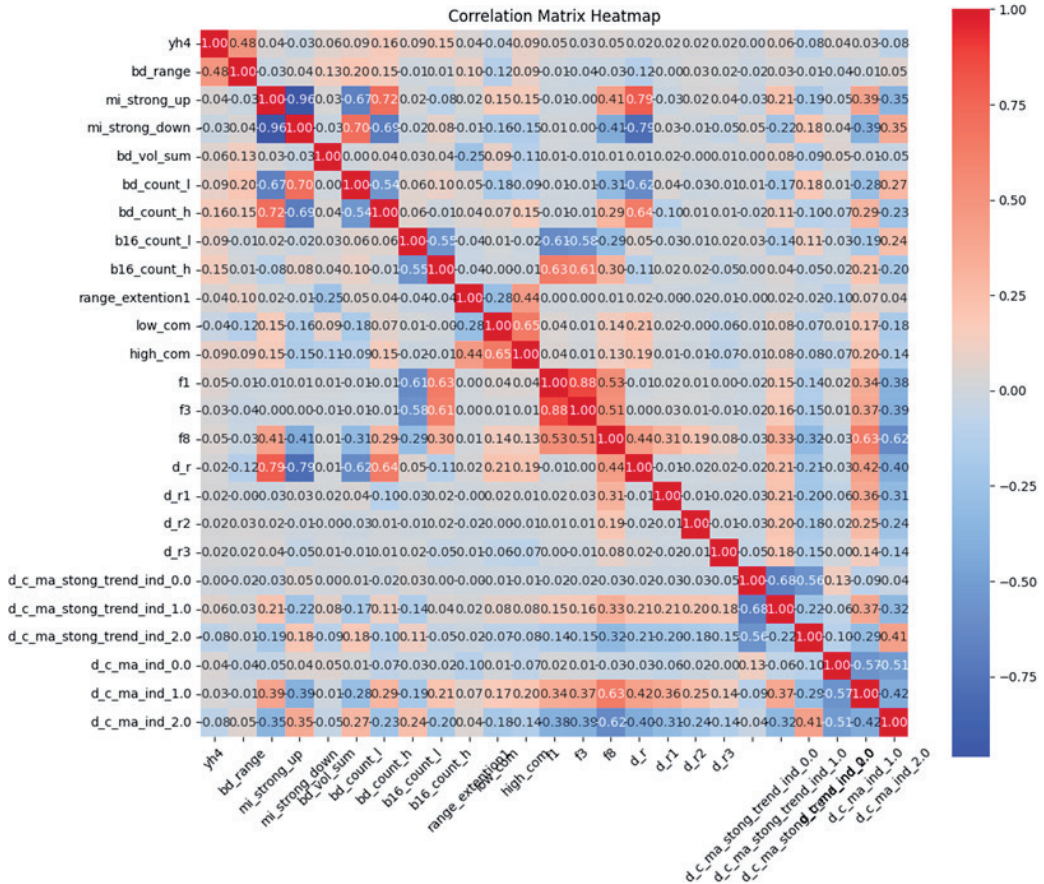
Анхны өгөгдлөөс тухайн өдөрт хамаарах минутуудаас бүрдсэн хугацаан цувааны онцлог шинж чанарыг тооцоолон өдрийн өгөгдөл рүү шилжүүлэхэд нийт 2,102 өдрийн боломжит өгөгдөл үүссэн. Өдрийн өгөгдлийг ашиглан дахин өдөр, 7 хоног, сарын шинж чанаруудыг ялган авахад нийт 176 төрлийн онцлог шинж чанарыг (feature) агуулсан 1,689 өдрийн өгөгдөл үүссэн. Хавсралт А.2.1, код хэрэгжүүлэлтэд тусгасан. А.2.3–д дурдсан

кодыг хэрэгжүүлэн ялгаж авсан. Нийт өгөгдлөөс python хэл дээр онцлог ялгах алгоритмыг хэрэгжүүлж Intel Core i7–12700K CPU, 24 GB RAM, Python 3.6.13 ашиглан тооцоолол хийхэд нийт 14 цаг тооцоологдсон. Тооцооллыг хийж гүйцэтгэхэд 24GB RAM хангалтгүй байсан тул өгөгдлийг Dataframe үүсгэн 3 хувааж тухайн өдөр харгалзах онцлог шинжүүдийг ялгаж авсны дараа dataframe–үүдийг нэгтгэв.

Нийт 176 өгөгдлөөс хавсралт А.2.4–д хамааралд үндэслэн дараах 22 онцлог шинж дурдсан кодыг хэрэгжүүлэн, корреляци чанаруудыг ялгаж авсан. Үүнд:

д/д	Хувьсагч	Тайлбар
1	<i>bd_range</i>	Тухайн өдрийн өмнөх өдрийн хамгийн дээд болон доод ханшийн зөрүү
2	<i>mi_strong_up</i>	Тухайн өдрийн буюу Токёо нээгдэхээс Нью-Йорк нээгдэх хүртэл нийт 1600 минутад 2, 4, 8 цагийн хөдлөх дундаж нэгэн чиглэлд өссөн минутын тоо
3	<i>mi_strong_down</i>	Тухайн өдрийн буюу Токёо нээгдэхээс Нью-Йорк нээгдэх хүртэл нийт 1600 минутад 2, 4, 8 цагийн хөдлөх дундаж нэгэн чиглэлд буурсан минутын тоо
4	<i>bd_vol_sum</i>	Өмнөх өдрийн арилжаасан лотны нийлбэр.
5	<i>bd_count_l</i>	Өмнөх өдөр нийт хэдэн удаа доод ханш тогтоосон тоо
6	<i>bd_count_h</i>	Өмнөх өдөр нийт хэдэн удаа дээд ханш тогтоосон тоо
7	<i>b16_count_l</i>	Тухайн өдөр нийт хэдэн удаа доод ханш тогтоосон тоо
8	<i>b16_count_h</i>	Тухайн өдөр нийт хэдэн удаа дээд ханш тогтоосон тоо
9	<i>d_c_ma_strong_trend_ind</i>	Тухайн өдрийн хаалтын ханшийн хөдлөх дундаж нь 5 ба 10 өдрийн хөдлөх дунджаар илүү эсвэл бага байсан
10	<i>range_extention1</i>	Өмнөх өдрийн Азийн ранжи болон нийт ранжи хэр зөрүүтэй байсан
11	<i>low_com</i>	Өмнөх өдрийн доод ханшнаас тухайн өдрийн доод ханшийн зөрүү ам.доллар
12	<i>high_com</i>	Өмнөх өдрийн дээд ханш тухайн өдрийн дээд ханшийн зөрүү ам.доллар
13	<i>f1</i>	тухайн өдрийн хаалт нээлтийн зөрүү нь тухайн өдрийн рэнжитэй харьцуулсан харьцаа
14	<i>f3</i>	Тухайн өдрийн хаалт нээлтийн зөрүү нь өмнөх өдрийн рэнжитэй харьцуулсан харьцаа
15	<i>f8</i>	Тухайн долоо хоногийн нээлт ба тухайн өдрийн хаалтын зөрүү нь тухайн долоо хоногийн рэнжитэй харьцуулсан харьцаа
16	<i>d_r</i>	Тухайн өдрийн Токёо болон Нью-Йорк нээгдэх үеийн ханшийн өсөлтийн хувь
17	<i>d_c_ma_ind</i>	Өмнөх өдрийн хаалтын ханшуудын хөдлөх дунджаас хэр зөрүүтэй яваа байдал
18	<i>d_r1, d_r2, d_r3</i>	Өсөлтийн хувийн 1, 2, 3 хоногоор хоцролт авсан хувьсагч
19	<i>p</i>	Нью-Йоркийн зах зээл нээгдэх үеийн ханш
20	<i>y</i>	Нью-Йоркийн зах зээл нээгдснээс хойш 4 цагийн дараах ханш

Зураг 12. Оролтын өгөгдлийн корреляцийн матриц



4.4 ӨГӨГДӨЛ БОЛОВСРУУЛАЛТ

Анхны өгөгдлөөс тухайн өдөрт хамаарах минутуудаас бүрдсэн хугацаан цувааны онцлог шинж чанарыг тооцоолон өдрийн өгөгдөл рүү шилжүүлэхэд нийт 2102 өдрийн боломжит өгөгдөл үүссэн. Өдрийн өгөгдлийг ашиглан дахин өдөр, 7 хоног, сарын шинж чанаруудыг ялган авахад нийт 176 төрлийн шинж чанарыг агуулсан 1,689 өдрийн өгөгдөл үүссэн. Гүн сургалтад ашиглах оновчтой шинж чанаруудыг сонгохдоо:

1. Оролтын өгөгдөл хоорондын collinearity хамаарал байгаа эсэхийг шалгаж корреляци 0.9 дээш утгуудтай шинж чанарууд хассан;
2. Оролтын өгөгдөл дотор вариацигүй буюу тогтмол өгөгдөл орсон эсэхийг шалгаж, вариаци нь 0.1 доош байвал гэсэн хязгаар тавьж хэрэггүй шинж чанаруудыг хассан;

3. Оролтын өгөгдөлд Principal component analysis 95 хувь байх хувиргалт хийсэн; Сургалтын ба туршилтын өгөгдөл: Загварыг үнэлэх, мөн үнэлгээний чанарыг шалгахын тулд өгөгдлийг хоёр хэсэгт хуваасан:

1. Сургалтын өгөгдөл: Нийт оролтын өгөгдлийн 80 хувь;
2. Тест хийх өгөгдөл: Нийт оролтын өгөгдлийн 20 хувь;

V. ЗАГВАР БА ТУРШИЛТ

5.1 ОРЧНЫ ТОХИРГОО

Алтны үнэ таамаглах загварыг хэрэгжүүлэх, туршлага хуримтлуулах нэг чухал зүйл нь машин сургалтын орчин юм. Бид техник болон өгөгдлийн боломждоо тулгуурлан дараах орчныг бэлдсэн ба цаашид хүчин чадлыг нэмэгдүүлэх хэрэгцээ байгаа.

- Компьютер: Intel Core i7-12700K CPU, 24GB RAM
- Програмчлалын хэл: Python 3.6.13
- Ашигласан сангууд:
- Өгөгдөл тооцоолох: Numpy, Pandas, Matplotlib, Seaborn,
- Загварын үнэлгээ: Sklearn (train_test_split, MinMaxScaler, StandardScaler, VarianceThreshold), tensorflow (Dense, Dropout, Flatten, Adam, SGD);
- Зураг дүрслэл: matplotlib (plt), seaborn, tensorboard

5.2 СУУРЬ ЗАГВАР (BASELINE MODEL)

Нейроны сүлжээний суурь загварыг үүсгэхдээ хайперпараметруудийг доорх байдлаар сонгов:

- Нууц давхаргад Relu идэвхжүүлэх функц;
- Гаралтын давхаргад Softmax функц;
- Adam optimizer(learning_rate=0.0001)
- Алдагдлын функцэд loss=Categorical_crossentropy, kernel_regularizer=L2(0.01)
- Нууц давхарга бүрд хаях коэффициент Dropout=0.2
- batch_size 16
- epochs 500.

Дээрх тохиргооны дагуу энгийн гүн сургалтын архитектур хэрэгжүүлж туршсан ба нууц давхаргын тоог өөрчилж гаралтын үр дүнг ажиглан судалсан.

Хүснэгт 2. Суурь загварын үнэлгээний утгууд

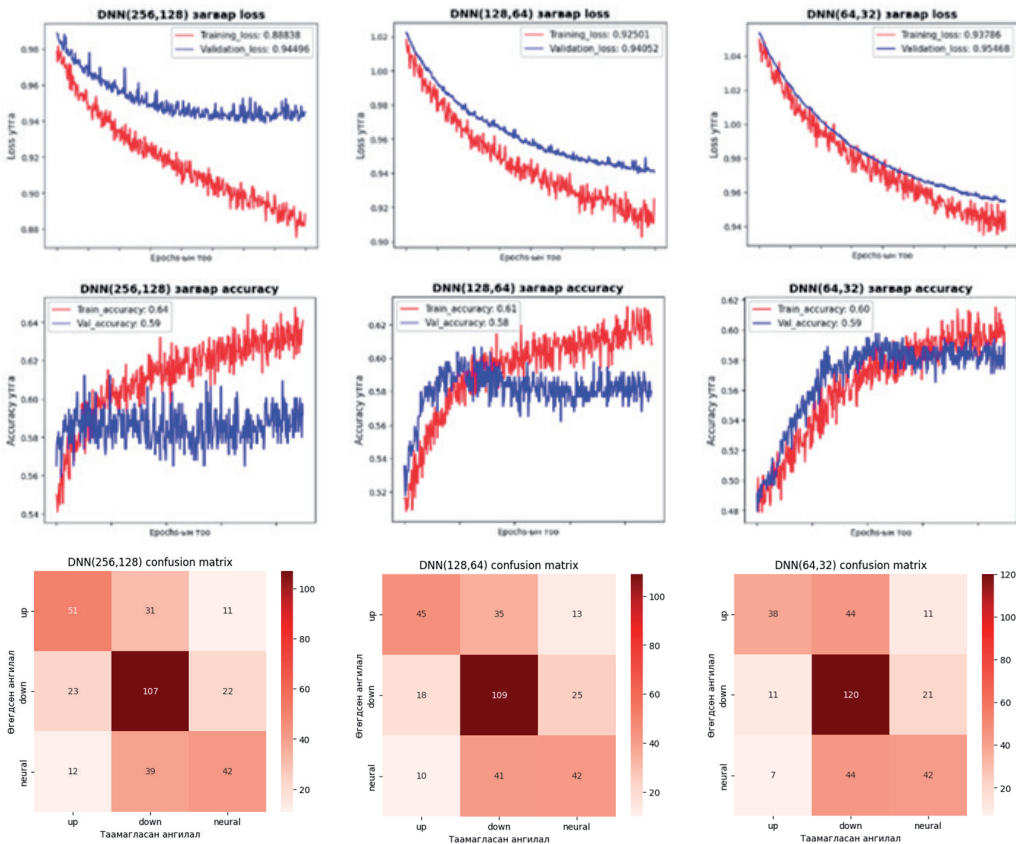
	Нууц давхарга	Параметр	Accuracy rate		Loss		
			Training	Validation	Training	Validation	Зөрүү
Загвар I	(256; 128)	72,707	64%	59%	0.88	0.94	0.06
Загвар II	(128; 64)	11,523	61%	58%	0.92	0.94	0.02
Загвар III	(64; 32)	3,715	60%	59%	0.93	0.95	0.02

Эх сурвалж: Судлаачийн тооцоолол, python, tensorflow

Загварын нууц давхаргад байрлах нейроны тоог өөрчлөх нь загварын Accuracy-д нөлөөлөхгүй боловч загварын Loss-д нөлөөтэй байна. DNN(64,32) загварын

сургалтын болон баталгаажуулах түүврийн алдааны утга бусад DNN(256,128), DNN(128,64) загваруудтай харьцуулахад илүү сайжирч, 0.02-оор зөрүү багассан байна.

Зураг 13. DNN загваруудын алдагдлын утга ба accuracy



Эх сурвалж: Судлаачийн тооцоолол, python, tensorflow

Үнэлэгдсэн DNN(64,32) загварт үнэ дээшээ өссөн утгын 41 хувийг зөв таамагласан, бол 47 хувьд үнэ буурна, 12 хувьд нь үнэд өөрчлөлт орохгүй гэж таамагласан бол үнэ буурах өгөгдлийн 79 хувийг буурна гэж зөв таамагласан, 7 хувийг өснө, 14 хувийг хэвэндээ байна гэж таамагласан. Үнэ хэвэндээ байх өгөгдлийн 45 хувийг зөв

таамагласан бол 8 хувийг өснө, 47 хувийг буурна гэж таамагласан байна.

Эндээс харахад үнэ буурах хөдөлгөөнийг маш сайн таамагласан боловч үнэ өсөх, хэвэндээ байх хөдөлгөөнийг муу таамагласан байна.

5.3 LSTM ЗАГВАРЫН ТУРШИЛТ

LSTM загварыг үүсгэхдээ хайперпараметруудийг доорх байдлаар сонгов.

- Нууц давхаргад Relu идэвхжүүлэх функц;
- Гаралтын давхаргад Dense 1 байхаар;
- Adam optimizer(learning_rate=0.005)
- Алдагдлын функцэд loss=Mean_
squared_error
- Нууц давхарга бүрд хаях коэффициент

Dropout =0.2

- batch_size 32
- epochs 50.
- validation_split = 0.1

Дээрх тохиргооны дагуу LSTM сургалтын архитектур хэрэгжүүлж туршсан ба нууц давхаргын тоог өөрчилж гаралтын үр дүнг ажиглан судалсан.

Хүснэгт 3. LSTM загварын үр дүн

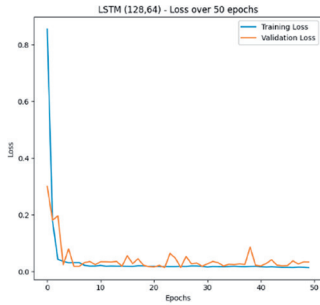
	Параметр	Mean squared error		Зөрүү	MSE	MAPE	MAE
		Training	Validation				
LSTM(256,128)	486,017	0.0118	0.0336	0.0218	7,833.89	3.99%	77.18
LSTM(128,64)	128,321	0.0120	0.0375	0.0255	7,160.50	3.75%	72.83
LSTM(64,32)	35,489	0.0188	0.0230	0.0042	5,136.74	3.06%	59.56

Эх сурвалж: Судлаачийн тооцоолол, python, tensorflow

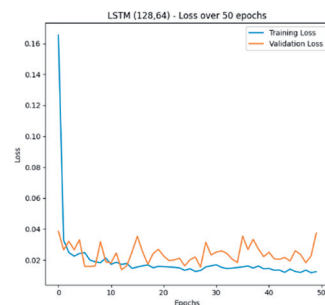
Загварын нууц давхаргад байрлах нейроны тоог өөрчлөх нь загварын таамаглах чадварт нөлөөтэй байна. LSTM(64,32) загварын сургалтын болон баталгаажуулах

MSE илэрхийлсэн таамаглах чадвар бусад LSTM(256,128), LSTM(128,64) загваруудтай харьцуулахад илүү сайжирсан байна.

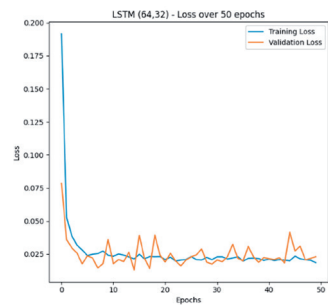
Зураг 14. (256,128) загварын Accuracy



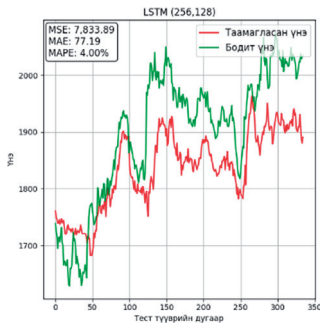
Зураг 15. (128,64) загварын Accuracy



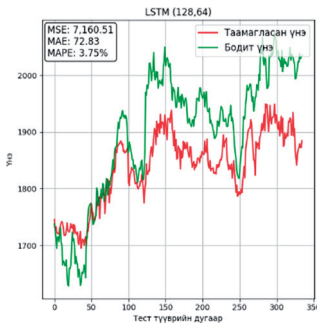
Зураг 16. (64,32) загварын Accuracy



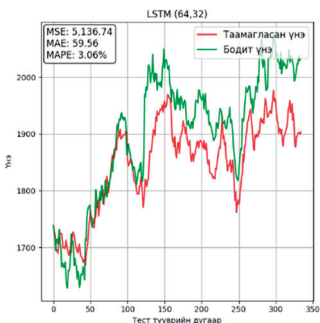
Зураг 17. (256,128) таамагласан утга



Зураг 18. (128,64) таамагласан утга



Зураг 19. (64,32) таамагласан утга



Үнэлэгдсэн LSTM(64,32) загвараар тест түүвэр дотор таамаглал дэвшүүлэн MSE утгыг олоход 5,136.74, MAPE 3.06%, MAE 59.56 байхаар тооцоологдов. Энэхүү судалгааны LSTM (64,32)-ын MAPE 3.06

гэж тооцоологдсон үр дүнг [6]-д дурдсан LSTM загваруудын MAPE-ээр хэмжсэн үр дүнтэй харьцуулахад алдаа өндөр гэж үзэж болохоор байна (Хүснэгт 4).

Хүснэгт 4. LSTM үр дүнг бусад судалгааны үр дүнтэй харьцуулсан хүснэгт

	MAPE /Mean absolute percentage error/
LSTM(64,32)	3.06
[35]	1.03
[6]	1.69
[40]	4.13
[41]	1.007
[57]	0.119
[60]	0.91
[66]	0.63

[61]*-д дурдсан үр дүн

Эх сурвалж: [61] судалгааны 17 дугаар хуудас

5.4 ЗАГВАРЫН ХАРЬЦУУЛАЛТ

Суурь загвар нь алтны үнийг тодорхой цэгээс дээшээ, доошоо, эсвэл $[-5.3, 5.3]$ интервалд байх тухай гурван таамаг дэвшүүлэн мадаглалыг тооцож байгаа. Харин LSTM загвар нь алтны үнэ яг хэд байх тухай $t+1$ үеийн утгийг таамаглаж байгаа. Тиймээс

хоёр загварыг харьцуулахын тулд LSTM загвараас гарч ирсэн $t+1$ үеийн таамаглалын үнийг $[-\infty, -5.3]$, $[-5.3, 5.3]$, $[5.3, +\infty]$ гэсэн гурван интервалд хуваан хэрхэн зөв буруу таамагласан талаар төөрөгдлийн матрицыг тооцов.

Хүснэгт 5. Загварын харьцуулалт

Таамагласан утга

		Up	Down	Neural	Төөрсөн хувь
Бодит утга	Up	Суурь - 41% LSTM - 29%	Суурь - 47% LSTM - 66%	Суурь - 12% LSTM - 5%	Суурь - 59% LSTM - 71%
	Down	Суурь - 7% LSTM - 28%	Суурь - 79% LSTM - 71%	Суурь - 14% LSTM - 1%	Суурь - 21% LSTM - 29%
	Neural	Суурь - 8% LSTM - 22%	Суурь - 47% LSTM - 73%	Суурь - 45% LSTM - 4%	Суурь - 55% LSTM - 96%

20% – иас дээш төөрсөн тохиолдлыг улаанаар,

5% – аас доош төөрсөн тохиолдлыг ногооноор,

50% – иас дээш зөв таамагласан тохиолдлыг ногооноор тус тус тэмдэглэв.

Эх сурвалж: Судлаачийн тооцоолол

Тооцоологдсон төөрөгдлийн матрицаас харахад буурсан тохиолдлыг суурь загварт 79 хувь, LSTM(64,32) загварт 71 хувьтай зөв таамагласан байна. Харин бусад тохиолдлуудад суурь загвар нь LSTM(64,32) загвараас харьцангуй сайн таамагласан байна.

LSTM(64,32) загварт таамаглалын төөрсөн тохиолдол суурь загвартай харьцуулахад

өндөр байна. Өөрөөр хэлбэл, хэвэндээ байх тохиолдлын 73 хувийг буурна, 22 хувийг өснө, 4 хувийг хэвэндээ байна гэж таамагласан байна. Өсөх тохиолдлын 66 хувийг буурна, 5 хувийг хэвэндээ байна, 29 хувийг өснө гэж тус тус таамагласан байна. Суурь загвар нь LSTM(64,32)–ээс илүү бага хувиар төөрсөн таамаглал дэвшүүлж байна.

ДҮГНЭЛТ

Судалгааны ажлаараа гүн сургалтын арга ашиглан нэг унц алтны үнэ өсөх, буурах, хэвэндээ байх гэсэн 3 ангиллаар богино хугацаанд таамаглал дэвшүүлэх боломжтой эсэхийг судаллаа. Гүн сургалтын загварт нийт 2,882,154 минутын өгөгдлөөс тухайн өдрийн минутын хугацаан цуваанаас хамгийн өндөр нөлөөтэй 22 шинж чанаруудыг ялган авч нийт 1,689 өдрийн тоог оролт өгөгдлөөр үүсгэсэн.

Үүсгэсэн өгөгдлийг ашиглан хоёр төрлийн загвар үүсгэж туршсан. Нэгдүгээрт энгийн гүн сургалтын арга ашиглан суурь загвар (baseline model) үүсгэсэн ба загварын нууц давхарга дахь нейроны тоог (256, 128)–аас (64, 32) болгож бууруулахад загварын сургалтын болон баталгаажуулах түүврийн алдааны утга 0.02–оор буурч, загварын үнэлгээ сайжирсан. Харин (64, 32) болгоход 60 хувийн магадлалтай зөв таамагласан байна. Оролтын өгөгдөлд үнэ буурах хөдөлгөөнийг илүү сайн таамагласан боловч үнэ өсөх, хэвэндээ байх хөдөлгөөнийг оролтын өгөгдлөөс 53–59 хувийн зөрүүтэй таамагласан байна.

Хоёрдугаарт цуваа өгөгдөлд хамгийн өндөр үр дүн үзүүлдэг LSTM загвар болон онцлогийн хамгийн чухлыг авах PCA аргатай хослуулан туршсан. Энэ загварын үр дүн нь LSTM загварын нейроны тоог (256,128)–аас (64,32) болгож бууруулахад MSE 7,833.89–өөс 5,136.74 болж буурсан, MAPE 3.99%–аас 3.06% болж буурсан, MAE 77.18–аас

59.56 болж тус тус буурсан байна.Энэхүү судалгаанд үнэлсэн LSTM загварын үр дүн [6]–д дурдсан LSTM загваруудын MAPE үр дүнтэй харьцуулахад таамаглалын алдаа өндөр байна.

Суурь загварыг LSTM загвартай харьцуулахад үнэ буурах тохиолдлыг 79 хувь, 71 хувьтай тус тус зөв таамаглаж байна. Үнэ өсөх нөхцөл байдлыг суурь загварт 59 хувь, LSTM загварт 71 хувьтай төөрсөн таамаглал дэвшүүлсэн бол үнэ хэвэндээ байх нөхцөл байдлыг суурь загварт 55 хувь, LSTM загварт 96 хувьтай төөрсөн таамаглал тус тус дэвшүүлж байна. Энэхүү судалгаанд тооцоолсон суурь загвар нь LSTM загвараас илүү бага хувиар төөрсөн таамаглал дэвшүүлж байна. LSTM загвараар суурь загварын таамагласан үр дүнг шалгах нь тохиромжгүй гэж үзэж байна.

Энэ туршилтыг цаашид гүн сургалтын загварын оролтын өгөгдлийн хэмжээг ихэсгэх, бусад эдийн засгийн ач холбогдолтой өндөр давтамжтай өгөгдлөөр өргөтгөх, LLM ашиглан зах зээлийн төлөв тодорхойлсон өгөгдлөөр өргөтгөх, мөн таамаглаж чадахгүй байгаа үлдсэн 40 хувь өгөгдлийн шинж чанар онцлогийг судлахаар зорьж байгаа болно.

Цаашид үнийн таамаглал дэвшүүлэх загварыг боловсронгуй болгосноор санхүүгийн эрсдэл, ашигт ажиллагаа зэрэг өргөн хэрэглээнд нэвтрүүлэх боломжтой юм.

АШИГЛАСАН МАТЕРИАЛ

- [1] “Introduction to Machine Learning with python”, 2017;
- [2] Arnuz Jentzen “Mathematical Introduction to Deep Learning Methods, Implementations and [1] Theory”, (2023); [\url{https://arxiv.org/pdf/2310.20360}](https://arxiv.org/pdf/2310.20360)
- [3] Krugman, Obstfeld and Melits. International economics: Theory and policy, 10th edition;
- [4] Nitin Indurkha and Fred J.Damerou. Hand book of Natural language processing, second edition;
- [5] Jiawei, Jiangtao “An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock” (2019);
- [6] Zexin Hu, Yiqi Zhao, Matloob Kushi, “A survey of forex and stock price prediction using deep learning” (2020);
- [7] Wataru Souma, Irena Vodenska, Hideaki Aoyama. “Enhanced news sentiment analysis using deep learning methods”. Journal of Computational Social Science (2019);
- [8] Xiao Zhong, David Enke. “Predicting the daily return direction of the stock market using hybrid machine learning. Springer open”(2019);
- [9] Sai Krishna, John McCrae. “A Comparative Study of SVM and LSTM Deep learning Algorithms for Stock market prediction” (2019);
- [10] Yoojeong song, Jae Won Lee, Jongwoo Lee, “A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction”(2018);
- [11] ArmanKhadjeh, SaeedAghabozorgi, TehYingWah. “Textminingofnews-headlinesforforexmarket prediction: Multi-layer dimension reduction algorithm with semantics and sentiment” (2014);
- [12] Grane Aurea, Velga Helena. “Wavelet-based detection of outliers in volatility models. Statistics and Econometrics”(2009);
- [13] Javen Qinfeng Shi, болон бусад, “Stock market prediction via Deep learning techniques: Survey” (2023);
- [14] Victor Chang “An innovative neural network approach for stock market prediction” 2018;
- [15] Frank c.park “Deep learning networks for stock market analysis and prediction: Methodology, Data Representation, and Case Studies”, (2017);
- [16] Lina Ni, Yujie Li, “Forecasting of forex time series data based on deep learning” 2018;
- [17] Sidra Mehtab, Jaydip Sen, Abhishek Dutta, “Stock price prediction using machine learning and LSTM-based deep learning models”, 2021;
- [18] Christopher Olah (2015) Understanding LSTM Networks [\url{http://colah.github.io/posts/2015-08-Understanding-LSTMs/}](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
- [19] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria (2018) Recent Trends in Deep Learning Based Natural Language Processing [\url{https://arxiv.org/abs/1708.02709}](https://arxiv.org/abs/1708.02709)

ХАВСРАЛТ

КОД ХЭРЭГЖҮҮЛЭЛТ

А.1 Түүхий өгөгдлийг нэг форматад оруулах

```
1 def DataReader(read_path):
2     df = pd.read_csv(read_path, delimiter= '\t' )
3     df = df.rename(columns={'<DATE>':'date', '<TIME>':'time', '<OPEN>':'
4         open', '<HIGH>':'high', '<LOW>':'low',
5             '<CLOSE>':'close', '<TICKVOL>':'tickvol', '<VOL
6                 >':'vol', '<SPREAD>':'spread'})
7     df['date'] =pd.to_datetime(df['date'], format='%Y.%m.%d')
8     df['date'] = df['date'].dt.strftime('%Y-%m-%d')
9     combined = df.date.str.cat(df.time, sep='_')
10    df['date_time'] = pd.to_datetime(combined)
11    df = df.loc[:, ~df.columns.str.contains('Unnamed')]
12    df.insert(0, 'date_time', df.pop('date_time'))
13    df.loc[(df['time'] > "00:00:00") & (df['time'] <= "04:00:00"), '
14        zone4'] = 1
15    df.loc[(df['time'] > "04:00:00") & (df['time'] <= "08:00:00"), '
16        zone4'] = 2
17    df.loc[(df['time'] > "08:00:00") & (df['time'] <= "12:00:00"), '
18        zone4'] = 3
19    df.loc[(df['time'] > "12:00:00") & (df['time'] <= "16:00:00"), '
20        zone4'] = 4
21    df.loc[(df['time'] > "16:00:00") & (df['time'] <= "20:00:00"), '
22        zone4'] = 5
23    df.loc[(df['time'] > "20:00:00") & (df['time'] <= "23:59:59"), '
24        zone4'] = 6
25    df['trade_day'] = df['date_time'].dt.dayofweek + 1
26    return df
```

А.2 Онцлог шинж чанар ялгахгаас өмнөх өгөгдлийн боловсруулалт

А.2.1 Зах зээл халалт, хөрөлтийг хэмжих параметр ялгах

```
1 #
2 df['mi_r'] = df.close.pct_change()*100
3 #
4 df['mi_g'] = df.close.diff()
5 #
6 df['mi_range'] = df.high - df.low
7 #
8 # 120,240
9 cnames = [ '120', '240' ]
10 for item in cnames:
11     df[f'mi_ma{item}'] = df['close'].rolling(window=int(item)).mean()
12 df = df.dropna()
13 df['over_heat_h2_in_minutes'] = df.close/df.mi_ma120
14 df['over_heat_h4_in_minutes'] = df.close/df.mi_ma240
```

A.1 Түүхий өгөгдлийг нэг форматад оруулах

```

1 def DataReader(read_path):
2     df = pd.read_csv(read_path, delimiter= '\t' )
3     df = df.rename(columns={'<DATE>':'date', '<TIME>':'time', '<OPEN>':'
4         open', '<HIGH>':'high', '<LOW>':'low',
5             '<CLOSE>':'close', '<TICKVOL>':'tickvol', '<VOL
6                 >':'vol', '<SPREAD>':'spread'})
7     df['date'] =pd.to_datetime(df['date'], format='%Y.%m.%d')
8     df['date'] = df['date'].dt.strftime('%Y-%m-%d')
9     combined = df.date.str.cat(df.time, sep='_')
10    df['date_time'] = pd.to_datetime(combined)
11    df = df.loc[:, ~df.columns.str.contains('Unnamed')]
12    df.insert(0, 'date_time', df.pop('date_time'))
13    df.loc[(df['time'] > "00:00:00") & (df['time'] <= "04:00:00"), '
14        zoneh4'] = 1
15    df.loc[(df['time'] > "04:00:00") & (df['time'] <= "08:00:00"), '
16        zoneh4'] = 2
17    df.loc[(df['time'] > "08:00:00") & (df['time'] <= "12:00:00"), '
18        zoneh4'] = 3
19    df.loc[(df['time'] > "12:00:00") & (df['time'] <= "16:00:00"), '
20        zoneh4'] = 4
21    df.loc[(df['time'] > "16:00:00") & (df['time'] <= "20:00:00"), '
22        zoneh4'] = 5
23    df.loc[(df['time'] > "20:00:00") & (df['time'] <= "23:59:59"), '
24        zoneh4'] = 6
25    df['trade_day'] = df['date_time'].dt.dayofweek + 1
26    return df

```

A.2 Онцлог шинж чанар ялгах аас өмнөх өгөгдлийн боловсруулалт

A.2.1 Зах зээл халалт, хөрөлтийг хэмжих параметр ялгах

```

1     #
2     df['mi_r'] = df.close.pct_change()*100
3     #
4     df['mi_g'] = df.close.diff()
5     #
6     df['mi_range'] = df.high - df.low
7     #
8     cnames = [ '120', '240' ]
9     for item in cnames:
10        df[f'mi_ma{item}'] = df['close'].rolling(window=int(item)).mean()
11    df = df.dropna()
12    df['over_heat_h2_in_minutes'] = df.close/df.mi_ma120
13    df['over_heat_h4_in_minutes'] = df.close/df.mi_ma240

```

A.1 Түүхий өгөгдлийг нэг форматад оруулах

```

1 def DataReader(read_path):
2     df = pd.read_csv(read_path, delimiter= '\t' )
3     df = df.rename(columns={'<DATE>':'date', '<TIME>':'time', '<OPEN>':'
         open', '<HIGH>':'high', '<LOW>':'low',
4         '<CLOSE>':'close', '<TICKVOL>':'tickvol', '<VOL
         >':'vol', '<SPREAD>':'spread'})
5     df['date'] =pd.to_datetime(df['date'], format='%Y.%m.%d')
6     df['date'] = df['date'].dt.strftime('%Y-%m-%d')
7     combined = df.date.str.cat(df.time, sep='_')
8     df['date_time'] = pd.to_datetime(combined)
9     df = df.loc[:, ~df.columns.str.contains('Unnamed')]
10    df.insert(0, 'date_time', df.pop('date_time'))
11    df.loc[(df['time'] > "00:00:00" & (df['time'] <= "04:00:00")), '
        zoneh4'] = 1
12    df.loc[(df['time'] > "04:00:00" & (df['time'] <= "08:00:00")), '
        zoneh4'] = 2
13    df.loc[(df['time'] > "08:00:00" & (df['time'] <= "12:00:00")), '
        zoneh4'] = 3
14    df.loc[(df['time'] > "12:00:00" & (df['time'] <= "16:00:00")), '
        zoneh4'] = 4
15    df.loc[(df['time'] > "16:00:00" & (df['time'] <= "20:00:00")), '
        zoneh4'] = 5
16    df.loc[(df['time'] > "20:00:00" & (df['time'] <= "23:59:59")), '
        zoneh4'] = 6
17    df['trade_day'] = df['date_time'].dt.dayofweek + 1
18    return df

```

A.2 Онцлог шинж чанар ялгахгаас өмнөх өгөгдлийн боловсруулалт

A.2.1 Зах зээл халалт, хөрөлтийг хэмжих параметр ялгах

```

1     #
2     df['mi_r'] = df.close.pct_change()*100
3     #
4     df['mi_g'] = df.close.diff()
5     #
6     df['mi_range'] = df.high - df.low
7     #         120,240
8     cnames = [ '120', '240']
9     for item in cnames:
10        df[f'mi_ma{item}'] = df['close'].rolling(window=int(item)).mean()
11    df = df.dropna()
12    df['over_heat_h2_in_minutes'] = df.close/df.mi_ma120
13    df['over_heat_h4_in_minutes'] = df.close/df.mi_ma240

```

A.1 Түүхий өгөгдлийг нэг форматад оруулах

```

1 def DataReader(read_path):
2     df = pd.read_csv(read_path, delimiter= '\t' )
3     df = df.rename(columns={'<DATE>':'date', '<TIME>':'time', '<OPEN>':'
4         open', '<HIGH>':'high', '<LOW>':'low',
5             '<CLOSE>':'close', '<TICKVOL>':'tickvol', '<VOL
6                 >':'vol', '<SPREAD>':'spread'})
7     df['date'] = pd.to_datetime(df['date'], format='%Y.%m.%d')
8     df['date'] = df['date'].dt.strftime('%Y-%m-%d')
9     combined = df.date.str.cat(df.time, sep='_')
10    df['date_time'] = pd.to_datetime(combined)
11    df = df.loc[:, ~df.columns.str.contains('Unnamed')]
12    df.insert(0, 'date_time', df.pop('date_time'))
13    df.loc[(df['time'] > "00:00:00") & (df['time'] <= "04:00:00"), '
14        zoneh4'] = 1
15    df.loc[(df['time'] > "04:00:00") & (df['time'] <= "08:00:00"), '
16        zoneh4'] = 2
17    df.loc[(df['time'] > "08:00:00") & (df['time'] <= "12:00:00"), '
18        zoneh4'] = 3
19    df.loc[(df['time'] > "12:00:00") & (df['time'] <= "16:00:00"), '
20        zoneh4'] = 4
21    df.loc[(df['time'] > "16:00:00") & (df['time'] <= "20:00:00"), '
22        zoneh4'] = 5
23    df.loc[(df['time'] > "20:00:00") & (df['time'] <= "23:59:59"), '
24        zoneh4'] = 6
25    df['trade_day'] = df['date_time'].dt.dayofweek + 1
26    return df

```

A.2 Онцлог шинж чанар ялгах аас өмнөх өгөгдлийн боловсруулалт

A.2.1 Зах зээл халалт, хөрөлтийг хэмжих параметр ялгах

```

1 #
2 df['mi_r'] = df.close.pct_change()*100
3 #
4 df['mi_g'] = df.close.diff()
5 #
6 df['mi_range'] = df.high - df.low
7 # 120,240
8 cnames = [ '120', '240' ]
9 for item in cnames:
10     df[f'mi_ma{item}'] = df['close'].rolling(window=int(item)).mean()
11 df = df.dropna()
12 df['over_heat_h2_in_minutes'] = df.close/df.mi_ma120
13 df['over_heat_h4_in_minutes'] = df.close/df.mi_ma240

```

```

119     'd_o':[d_o], 'd_c':[d_c], 'd_h':[d_h], 'd_l':[d_l], 'd_rate':[
        d_rate], 'd_growth':[d_growth], 'd_range':[d_range], '
        mi_strong_up16':[mi_strong_up16], 'mi_strong_down16':[
        mi_strong_down16],
120     'd_tickvol_sum':[d_tickvol_sum], 'd_mi_tickvol_std':[
        d_mi_tickvol_std], 'd_vol_sum':[d_vol_sum], 'd_mi_vol_std':[
        d_mi_vol_std], 'd_mi_r_max':[d_mi_r_max], 'd_mi_r_mean':[
        d_mi_r_mean],
121     'd_mi_r_std':[d_mi_r_std], 'd_mi_r_var':[d_mi_r_var], '
        d_mi_range_max':[d_mi_range_max], 'd_mi_range_mean':[
        d_mi_range_mean], 'd_mi_range_std':[d_mi_range_std], '
        d_mi_range_var':[d_mi_range_var],
122     'y30':[y30], 'yh1':[yh1], 'yh2':[yh2], 'yh3':[yh3], 'yh4':[yh4], '
        yh5':[yh5], 'yh6':[yh6], 'yh7':[yh7], 'yh8':[yh8]
        }
123     new_data = pd.DataFrame(data)
124     model_data = pd.concat([model_data, new_data], ignore_index=
        True)
125
126     except TypeError:
127         print('TE:', item)
128         type_e_date.append(item)
129     except IndexError:
130         print('IE:', item)
131         index_e_date.append(item)
132     continue

```

A.2.3 Оцлог шинж чанаруудыг ялгах авах

```

1 def Ground_calculation(df):
2     keys1 = ['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7', 'p8']
3     keys2 = ['yh1', 'yh2', 'yh3', 'yh4', 'yh5', 'yh6', 'yh7', 'yh8']
4     for item in range(len(keys1)):
5         df[keys1[item]] = df[keys2[item]] - df['d_c']
6         p30 = ['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7', 'p8']
7         y30 = ['y1', 'y2', 'y3', 'y4', 'y5', 'y6', 'y7', 'y8']
8         for item in range(len(p30)):
9             iq33 = np.percentile(df[p30[item]], 33)
10            iq66 = np.percentile(df[p30[item]], 66)
11            for aa in range(len(df)):
12                if df[p30[item]][aa] > iq66:
13                    df.at[aa, y30[item]] = 'up'
14                elif iq33 <= df[p30[item]][aa] <= iq66:
15                    df.at[aa, y30[item]] = 'neural'
16                else:
17                    df.at[aa, y30[item]] = 'down'
18            return df
19 df = Ground_calculation(df)
20 cnames = ['5', '10', '50', '110']
21 for item in cnames:
22     df[f'd_vol_sum_ma{item}'] = df['d_vol_sum'].rolling(window=int(item))
        .mean()

```

```

23 df[f'd_tickvol_sum_ma{item}'] = df['d_tickvol_sum'].rolling(window=
    int(item)).mean()
24 df[f'd_o_ma{item}'] = df['bd_o'].rolling(window=int(item)).mean()
25 df[f'd_h_ma{item}'] = df['bd_h'].rolling(window=int(item)).mean()
26 df[f'd_l_ma{item}'] = df['bd_l'].rolling(window=int(item)).mean()
27 df[f'd_c_ma{item}'] = df['bd_c'].rolling(window=int(item)).mean()
28 df[f'd_mi_r_std_ma{item}'] = df['bd_mi_r_std'].rolling(window=int(
    item)).mean()
29 df[f'd_mi_range_std_ma{item}'] = df['bd_mi_range_std'].rolling(window
    =int(item)).mean()
30 cnames = ['5', '10', '50', '110']
31 for i in cnames:
32     for item in range(len(df)):
33         # daily calculation
34         if df[f'over_heat{i}'][item] > np.percentile(df[f'over_heat{i}'],
35             75) :
36             df.at[item, f'over_heat_ind{i}'] = 1
37         else:
38             df.at[item, f'over_heat_ind{i}'] = 0
39         if df[f'over_heat{i}'][item] < np.percentile(df[f'over_heat{i}'],
40             25) :
41             df.at[item, f'over_stag_ind{i}'] = 1
42         else:
43             df.at[item, f'over_stag_ind{i}'] = 0
44     cnames = ['5', '10', '50', '110']
45     for i in cnames:
46         for item in range(len(df)):
47             # daily calculation
48             if df[f'd_tickvol_sum_ma{i}'][item] > np.percentile(df[f'
49                 d_tickvol_sum_ma{i}'], 75) :
50                 df.at[item, f'over_active_tickvol_ind{i}'] = 1
51             else:
52                 df.at[item, f'over_active_tickvol_ind{i}'] = 0
53             if df[f'd_tickvol_sum_ma{i}'][item] < np.percentile(df[f'
54                 d_tickvol_sum_ma{i}'], 25) :
55                 df.at[item, f'over_inactive_tickvol_ind{i}'] = 1
56             else:
57                 df.at[item, f'over_inactive_tickvol_ind{i}'] = 0
58     for item in range(len(df)):
59         # daily calculation
60         if df['d_c_ma50'][item] < df['d_c_ma10'][item] < df['d_c_ma5'][item]
61             < df['d_c'][item]:
62             df.at[item, 'd_c_ma_strong_trend_ind'] = 1
63         elif df['d_c_ma50'][item] > df['d_c_ma10'][item] > df['d_c_ma5'][item]
64             > df['d_c'][item]:
65             df.at[item, 'd_c_ma_strong_trend_ind'] = 2
66         else:
67             df.at[item, 'd_c_ma_strong_trend_ind'] = 0
68     for item in range(len(df)):
69         # ===== daily features
70         =====
71         df.at[item, 'f1'] = (df['d_c'][item] - df['d_o'][item])/df['d_range'

```

```

104     df.at[item, 'd_mi_r_std_instable'] = 1
105     else:
106         df.at[item, 'd_mi_r_std_instable'] = 0

```

A.2.4 Корреляци хамаарал байгаа эсэхийг шалгаж, хэрэггүй онцлог шинжүүдийг хасах

```

1 redundant = []
2 for i, col in enumerate(df.corr().columns):
3     for j in range(0,i):
4         if(df.corr().iloc[j][col] >0.9 and col not in redundant):
5             redundant.append(col)
6 from sklearn.feature_selection import VarianceThreshold
7 sel = VarianceThreshold(threshold=0.1)
8 sel.fit(dx)
9 # Get indices of selected features
10 selected_indices = sel.get_support(indices=True)
11 # Get the names of the selected columns
12 selected_columns = dx.columns[selected_indices]
13 print("Selected_columns:")
14 print(selected_columns)

```

A.2.5 DNN загварын үнэлгээ

```

1 nn_columns = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, '
2     d_c_ma_stong_trend_ind_0.0', 'd_c_ma_stong_trend_ind_1.0',
3     'd_c_ma_stong_trend_ind_2.0', 'd_c_ma_ind_0.0', 'd_c_ma_ind_1
4     .0', 'd_c_ma_ind_2.0','y4']
5 pre_data = final_df[nn_columns]
6 part1 = final_df[final_df['y4']=='down']
7 part2 = final_df[final_df['y4']=='up']
8 part3 = final_df[final_df['y4']=='neural']
9 part3_sample = part3.sample(n=470, random_state=42)
10 part3_not_included = part3[~part3.index.isin(part3_sample.index)]
11 model_nn_data = pd.concat([part1, part2, part3_sample], ignore_index=
12     True)
13 nn_columns = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, '
14     d_c_ma_stong_trend_ind_0.0', 'd_c_ma_stong_trend_ind_1.0',
15     'd_c_ma_stong_trend_ind_2.0', 'd_c_ma_ind_0.0', '
16     d_c_ma_ind_1.0', 'd_c_ma_ind_2.0']
17 ground_thruth_nn = ['y4']
18 nn_x = model_nn_data[nn_columns]
19 nn_y = pd.get_dummies(model_nn_data[ground_thruth_nn], prefix='y4').
20     astype(int)
21 X = np.array(nn_x)
22 Y = np.array(nn_y)
23 keys = np.array(range(X.shape[0]))
24 np.random.shuffle(keys)
25 X = X[keys]
26 Y = Y[keys]

```

```

21 x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size
    =0.2, random_state=42)
22 model = Sequential()
23 model.add(Dense(units = 256, activation='relu', input_shape=(19,),
    kernel_regularizer=regularizers.l2(0.01)))
24 model.add(Dropout(0.2))
25 model.add(Dense(units = 128, activation='relu', kernel_regularizer=
    regularizers.l2(0.01)))
26 model.add(Dropout(0.2))
27 model.add(Dense(units = 64, activation='relu', kernel_regularizer=
    regularizers.l2(0.01)))
28 model.add(Dropout(0.2))
29 model.add(Dense(units = 3, activation='softmax'))
30 model.compile(optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
31             metrics=['accuracy'])
32 model.summary()
33 history = model.fit(x_train, y_train, epochs=300, batch_size = 16,
    validation_data=(x_test, y_test))
34 for i in range(x_test.shape[0]):
35     print(model.predict(x_test[i:i+1]))
36     print(y_test[i])

```

A.2.6 LSTM загварын үнэлгээ

```

1 columns_for_model_data = ['yh4', 'bd_range', 'mi_strong_up', '
    mi_strong_down', 'bd_vol_sum', 'bd_count_l', 'bd_count_h', '
    b16_count_l', 'b16_count_h',
2     'd_c_ma_strong_trend_ind', 'range_extention1', 'low_com', '
    high_com', 'f1', 'f3', 'f8', 'd_r', 'd_c_ma_ind', 'd_r1', '
    d_r2', 'd_r3']
3 data = df[columns_for_model_data]
4 d_c_ma_stong_trend_ind_dummies = pd.get_dummies(data['
    d_c_ma_strong_trend_ind'], prefix='d_c_ma_stong_trend_ind').astype(
    int)
5 d_c_ma_ind_dummies = pd.get_dummies(data['d_c_ma_ind'], prefix='
    d_c_ma_ind').astype(int)
6 data = pd.concat([data, d_c_ma_stong_trend_ind_dummies], axis=1)
7 data = pd.concat([data, d_c_ma_ind_dummies], axis=1)
8 data.drop(columns=['d_c_ma_strong_trend_ind', 'd_c_ma_ind'], inplace=
    True)
9 scaler = StandardScaler()
10 columns_to_scaler = ['yh4', 'bd_range', 'mi_strong_up', 'mi_strong_down'
    , 'bd_vol_sum', 'bd_count_l', 'bd_count_h', 'b16_count_l', '
    b16_count_h',
11     'range_extention1', 'low_com', 'high_com', 'f1',
    'f3', 'f8', 'd_r', 'd_r1', 'd_r2', 'd_r3']
12 columns_left = ['d_c_ma_stong_trend_ind_0.0', 'd_c_ma_stong_trend_ind_1
    .0', 'd_c_ma_stong_trend_ind_2.0', 'd_c_ma_ind_0.0', 'd_c_ma_ind_1.0
    ', 'd_c_ma_ind_2.0']
13 scaled_data = scaler.fit_transform(data[columns_to_scaler])

```

```

14 pca = PCA(95)
15 pca_data = pca.fit_transform(scaled_data)
16 pca_df = pd.DataFrame(pca_data, index=data.index, columns=[f'PC{i+1}'
17     for i in range(n_components)])
18 columns_left = ['d_c_ma_stong_trend_ind_0.0', 'd_c_ma_stong_trend_ind_1
19     .0', 'd_c_ma_stong_trend_ind_2.0',
20     'd_c_ma_ind_0.0', 'd_c_ma_ind_1.0', 'd_c_ma_ind_2.0']
21 final_df = pd.concat([pca_df, data[columns_left]], axis=1)
22
23 def create_sequences(data, sequence_length=14):
24     X, y = [], []
25     data_array = data.values
26     for i in range(14, len(data_array)):
27         X.append(data_array[i-sequence_length:i])
28         y.append(data_array[i, 0]) # 0 is the 'price' column
29     return np.array(X), np.array(y)
30 X, y = create_sequences(final_df, 14)
31 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
32     =0.2, shuffle=False)
33
34 learning_rate = 0.005
35 optimizer = Adam(learning_rate=learning_rate)
36 model = Sequential([
37     LSTM(64, activation='relu', return_sequences=True, input_shape=(
38         X_train.shape[1], X_train.shape[2])),
39     Dropout(0.2),
40     LSTM(32, activation='relu', return_sequences=False),
41     Dropout(0.2),
42     Dense(1)
43 ])
44 model.compile(optimizer=optimizer, loss='mean_squared_error', metrics
45     =['accuracy'])
46 model.summary()
47 history = model.fit(X_train, y_train, epochs=50, batch_size=32,
48     validation_split=0.1)
49 predicted_pca = model.predict(X_test) # shape: (n_samples, 1), in PCA
50     space (PC1)
51 n_samples = predicted_pca.shape[0]
52 n_pca_components = pca.n_components_
53 pca_full = np.zeros((n_samples, n_pca_components))
54 pca_full[:, 0] = predicted_pca.ravel() # Assume PC1 was the target
55 reconstructed_scaled = pca.inverse_transform(pca_full)
56 n_samples = predicted_prices.shape[0]
57 n_features = len(columns_to_scaler)
58 padded_predictions[:, :] = reconstructed_scaled
59 idx = columns_to_scaler.index('yh4')
60 padded_predictions[:, idx] = predicted_prices.ravel()
61 y_pred_rescaled = scaler.inverse_transform(padded_predictions[:, 0])
62 padded_y_test = np.zeros((n_samples, n_features))
63 padded_y_test[:, 0] = y_test.ravel()
64 y_test_original = scaler.inverse_transform(padded_y_test[:, 0])
65 df0 = pd.DataFrame(y_pred_rescaled, columns=['predicted'])

```

```

14 pca = PCA(95)
15 pca_data = pca.fit_transform(scaled_data)
16 pca_df = pd.DataFrame(pca_data, index=data.index, columns=[f'PC{i+1}'
17     for i in range(n_components)])
17 columns_left = ['d_c_ma_stong_trend_ind_0.0', 'd_c_ma_stong_trend_ind_1
18     .0', 'd_c_ma_stong_trend_ind_2.0',
19     'd_c_ma_ind_0.0', 'd_c_ma_ind_1.0', 'd_c_ma_ind_2.0']
19 final_df = pd.concat([pca_df, data[columns_left]], axis=1)
20
21 def create_sequences(data, sequence_length=14):
22     X, y = [], []
23     data_array = data.values
24     for i in range(14, len(data_array)):
25         X.append(data_array[i-sequence_length:i])
26         y.append(data_array[i, 0]) # 0 is the 'price' column
27     return np.array(X), np.array(y)
28 X, y = create_sequences(final_df, 14)
29 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
30     =0.2, shuffle=False)
31
32 learning_rate = 0.005
33 optimizer = Adam(learning_rate=learning_rate)
34 model = Sequential([
35     LSTM(64, activation='relu', return_sequences=True, input_shape=(
36         X_train.shape[1], X_train.shape[2])),
37     Dropout(0.2),
38     LSTM(32, activation='relu', return_sequences=False),
39     Dropout(0.2),
40     Dense(1)
41 ])
42 model.compile(optimizer= optimizer, loss='mean_squared_error', metrics
43     =['accuracy'])
44 model.summary()
45 history = model.fit(X_train, y_train, epochs=50, batch_size=32,
46     validation_split=0.1)
47 predicted_pca = model.predict(X_test) # shape: (n_samples, 1), in PCA
48     space (PC1)
49 n_samples = predicted_pca.shape[0]
50 n_pca_components = pca.n_components_
51 pca_full = np.zeros((n_samples, n_pca_components))
52 pca_full[:, 0] = predicted_pca.ravel() # Assume PC1 was the target
53 reconstructed_scaled = pca.inverse_transform(pca_full)
54 n_samples = predicted_prices.shape[0]
55 n_features = len(columns_to_scaler)
56 padded_predictions[:, :] = reconstructed_scaled
57 idx = columns_to_scaler.index('yh4')
58 padded_predictions[:, idx] = predicted_prices.ravel()
59 y_pred_rescaled = scaler.inverse_transform(padded_predictions[:, 0])
60 padded_y_test = np.zeros((n_samples, n_features))
61 padded_y_test[:, 0] = y_test.ravel()
62 y_test_original = scaler.inverse_transform(padded_y_test[:, 0])
63 df0 = pd.DataFrame(y_pred_rescaled, columns=['predicted'])

```